NPS-MA-97-004

NAVAL POSTGRADUATE SCHOOL

Monterey, California

SHOWCASE OF SUBROUTINES IN TRUE BASIC

BY

Allen V. Hershey

April 1997

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School
Monterey, CA 93943-5000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1997

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Showcase of Subroutines in True Basic

5. FUNDING NUMBERS

6. AUTHOR(S)
Allen V. Hershey

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-MA-97-004

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

On the mainframe there is a library of subroutines for mathematical physics. Many of the subroutines have been translated from FORTRAN into classic BASIC. They make a showcase of subroutines which compute special functions, interpolate data, and do matrix arithmetic. The showcase is under the control of True BASIC on a Macintosh Classic II computer from which it can be distributed on disk to other computers.

14. SUBJECT TERMS
BASIC, computer, special functions, interpolation, matrix arithmetic.

15. NUMBER OF PAGES
144

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT

NPS-MA-97-004 April 1997

SHOWCASE OF SUBROUTINES
in
True BASIC

By

ALLEN V. HERSHEY

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA 93943

Approved for public release; distribution unlimited.

TABLE OF CONTENTS

ABSTRACT

On the mainframe there is a library of subroutines for mathematical physics. Many of the subroutines have been translated from FORTRAN into classic BASIC. They make a showcase of subroutines which compute special functions, interpolate data, and do matrix arithmetic. The showcase is under the control of True BASIC on a Macintosh Classic II computer from which it can be distributed on disk to other computers.

INTRODUCTION

Where the same operations are applied to many different sets of data the operations can be executed by a subroutine which can be called from any place in the program. There is a library of subroutines on the main frame where the subroutines are written in FORTRAN. With the proliferation of home computers many of the subroutines have been translated into BASIC. Cartographic and Typographic subroutines have been converted into programs which have been documented in a previous report. Subroutines which compute special functions, interpolate data, and perform matrix operations are the subject of the present report.

The library has general purpose subroutines of high accuracy. Efficiency is not relevant to a home computer which is shut down every night, and without adequate accuracy a computation could be useless. The library is a showcase of classic programming.

There is Hewlett-Packard BASIC on an HP-85 computer, with Microsoft BASIC, Quick BASIC, and True BASIC on a Macintosh Classic II computer. There are Power Macintosh computers at KINKOS, and there are Macintosh Quadra computers at the Naval Postgraduate School. However, the Power Macintosh and the Macintosh Quadra computers will accept only True BASIC.

A test of the accuracy of computation is provided by the program:

```
1 PRINT USING "+#.###############^^^^": EXP(LOG(7))
2 END
```

When this test is tried on a computer, the numbers are as follows:

for HP BASIC on HP-85
```
   +7.00000 00000 30000E+000
```

for Quick BASIC on Classic II Macintosh
```
   +7.00000 00000 00200D+00
```

for True BASIC on Classic II Macintosh
```
   +6.99999 99998 64670e+00
```

for True BASIC on Power Macintosh
```
   +6.99999 99998 64670e+00
```

for True BASIC on Macintosh Quadra 700
```
   +7.00000 00000 00000e+00
```

A couple of these numbers are especially noisy. There may be differences because programming or processor may be single precision or double precision. That True BASIC does better in a computer with a coprocessor has been explained by Professor Kurtz.

Each subroutine in the showcase includes a main program which applies any test to the subroutine. Each subroutine is programmed in classic BASIC with each line numbered with a line number. Then BRANCH is possible. True BASIC does not tolerate line numbers unless they are consecutive. The subroutine is therefore at the top of the program where line numbers are immune, and the main program is at the bottom of the program where changes in line number are tolerable. The subroutines have been checked for self consistency or have been checked against tabulations by Watson or by Abramowitz and Stegun. Information about background is in the Bibliography.

Listings of the subroutines are given in Appendix A. Each page of each listing is flagged with the name of its subroutine.

MATHEMATICAL FUNCTIONS

Although True BASIC can perform the operations +−*/ with fourteen digit accuracy, the mathematical functions are limited to only eleven digit accuracy.
The full fourteen digit value of the exponential function is obtained by reference to the call line

CALL EXPN (X,Y)

whence Y is replaced by the exponential function of X.
The full fourteen digit value of the logarithmic function is obtained by reference to the call line

CALL LOGM (X,Y)

whence Y is replaced by the logarithmic function of X.

The accuracy test gives the result

EXPN(LOGM(7)) = +6.99999 99999 99920e+00

Although not one digit is correct in this number, the relative error is only one unit in ten to the fourteenth.

SPECIAL FUNCTIONS

Special functions have ascending power series which are absolutely convergent. However, if the argument is larger than one, the terms of the series overpower the sum and the series is lost in rounding error. Special functions have asymptotic series, but the truncation error in the sum increases with decrease in argument. There is a gap which cannot be reached by either series. Within the gap the function can be evaluated with full accuracy by a rational approximation which is valid in a zone in the complex plane. The zone is bounded by two concentric circles and the imaginary axis. As long as the roots of the denominator lie outside the zone, the function is analytic. If the accuracy of the rational approximation is bounded on the perimeter of the zone, then the accuracy of the rational approximation is even better within the zone.

One night there was a big snow storm in Dahlgren, and the computer would have been idle, had it not been given the task of computing rational approximations for a variety of special functions. The approximations were carried to sixteen digit accuracy.

ELLIPTIC INTEGRALS

The elliptic integral of the first kind is defined by the equation

$$F(\phi,k) = \int_0^\phi \frac{d\theta}{\sqrt{1 - k^2\sin^2\theta}}$$

and the elliptic integral of the second kind is defined by the equation

$$E(\phi,k) = \int_0^\phi \sqrt{1 - k^2\sin^2\theta}\, d\theta$$

where ϕ is the arc and k is the modulus. The elliptic integrals are returned by the call line

CALL ELLPCI (A,K,F,E)

where A is the arc, K is the modulus, F is the integral of the first kind, and E is the integral of the second kind. The integrals are computed with the Landen Transformation and with the DiDonato recurrence in different ranges of argument.

BESSEL FUNCTIONS

The solutions of the differential equation

$$\frac{d^2w}{dz^2}+\frac{1}{z}\frac{dw}{dz}+\left(1-\frac{n^2}{z^2}\right)w=0$$

are the ordinary Bessel functions $J_n(z)$ and $Y_n(z)$.

The function $J_n(z)$ is returned by reference to the call line

CALL BSSLJ (Z(),N,J())

and the function $Y_n(z)$ is returned by reference to the call line

CALL BSSLY (Z(),N,Y())

The order N is an unlimited integer.

The solutions of the differential equation

$$\frac{d^2w}{dz^2}+\frac{1}{z}\frac{dw}{dz}+\left(1+\frac{n^2}{z^2}\right)w=0$$

are the modified Bessel functions $I_n(z)$ and $K_n(z)$.

The function of the first kind is returned by reference to the call line

CALL BSSLI (M,Z(),N,I())

When the mode of operation M is 0 the subroutine returns $I_n(z)$, and when the mode of operation M is 1 the subroutine returns $e^{-z}I_n(z)$.

The function of the second kind is returned by reference to the call line

CALL BSSLK (M,Z(),N,K())

When the mode of operation M is 0 the subroutine returns $K_n(z)$, and when the mode of operation M is 1 the subroutine returns $e^{+z}K_n(z)$.

COMPLEX BESSEL FUNCTION

The Bessel function $J_\nu(z)$ is defined by the equation

$$J_\nu(z)=\sum_{m=0}^{\infty}\frac{(-1)^m(\frac{1}{2}z)^{\nu+2m}}{m!\Gamma(1+m+\nu)}$$

where the series is absolutely convergent for complex order and complex argument.

For small argument the function is computed with the convergent ascending series, while for large argument the function is computed by an asymptotic descending series. Between these two series the function is computed with the Debye approximation for which coefficients are given by Amos recurrence equations.

The function $J_\nu(z)$ is obtained by reference to the call line

CALL CBSSLJ (Z(),N(),J())

where Z is the complex argument, N is the complex order, and J is the complex function. In the limit as ν approaches an integer, the function is given by the subroutine BSSLJ.

The Bessel function $Y_\nu(z)$ is defined by the equation

$$Y_\nu(z)=\frac{J_\nu(z)\cos(\nu\pi)-J_{-\nu}(z)}{\sin(\nu\pi)}$$

except in the limit as ν approaches an integer, in which case the function is given by the subroutine BSSLY.

The Bessel function $I_\nu(z)$ is defined by the equation

$$I_\nu(z) = e^{-\nu\frac{1}{2}\pi i}J_\nu(e^{\frac{1}{2}\pi i}z)$$

where the order ν is unlimited.

The Bessel function $K_\nu(z)$ is defined by the equation

$$K_\nu(z) = \frac{1}{2}\pi\frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}$$

except in the limit as ν approaches an integer, in which case the function is given by the subroutine BSSLK.

The function K_ν is expressed in terms of $J_{-\nu}$ and J_ν. If ν is one third of an integer, then the accuracy of computation is excellent when the argument is less than one, but the accuracy deteriorates from rounding error when the argument increases above one. The functions

$$K_0, K_{\frac{1}{3}}, K_{\frac{2}{3}}, K_1$$

have rational approximations.

The function K_ν(z) is obtained by reference to the call line

CALL QBSSLK (Z(),N,K())

where Z is the complex argument, N is one third of any integer, and K is the complex function of order N.

The Airy function $Ai(z)$ is expressed by the equation

$$Ai(z) = +\frac{z^{\frac{1}{2}}}{\pi\sqrt{3}}K_{\frac{1}{3}}(\tfrac{2}{3}z^{\frac{3}{2}})$$

and its derivative $Ai'(z)$ is expressed by the equation

$$Ai'(z) = -\frac{z}{\pi\sqrt{3}}K_{\frac{2}{3}}(\tfrac{2}{3}z^{\frac{3}{2}})$$

Computations by subroutine QBSSLK are confirmed by tabulations of the Airy function.

GAMMA FUNCTION

The gamma function $\Gamma(z)$ is defined by the equation

$$\Gamma(z) = \frac{e^{-\gamma z}}{z}\prod_{n=1}^{\infty}\left[\frac{e^{\frac{z}{n}}}{1+\frac{z}{n}}\right]$$

where γ is Euler's constant. It is returned by reference to the call line

CALL CGAMMA (M,Z(),G())

where M is the mode of operation, Z is the argument, and G is the function. If M=0 the subroutine returns $\Gamma(z)$, while if M=1 the subroutine returns $\log\Gamma(z)$.

An important property of the gamma function is expressed by the equation

$$\Gamma(1+z) = z\Gamma(z)$$

The computation of the gamma function does not require a rational approximation.

EXPONENTIAL INTEGRAL

The complex exponential integral $Ei(z)$ of argument z is defined by the equation

$$Ei(z) = \int_{-\infty}^{z} \frac{e^t}{t}\, dt$$

where the path of integration lies in that part of the complex plane from which the positive real axis is excluded. The exponential integral is returned by reference to the call line

CALL EXPLI (M,Z(),F())

If the mode of operation M is 0 then the subroutine returns the function $Ei(z)$, but if the mode of operation M is 1 then the subroutine returns the function $e^{-z}Ei(z)$.

The complex function $Xi(z)$ of argument z is defined by the equation

$$Xi(z) = \int_{-\infty}^{z} \frac{e^t}{t^2}\, dt$$

Integration by parts shows that the function is given by the equation

$$\int_{-\infty}^{z} \frac{e^t}{t^2}\, dt = \int_{-\infty}^{z} \frac{e^t}{t}\, dt - \frac{e^z}{z}$$

The risk of roundoff error is reduced by cancellation. The function is returned by reference to the call line

CALL XPNLI (M,Z(),F())

If the mode of operation M is 0 then the subroutine returns the function $Xi(z)$, but if the mode of operation M is 1 then the subroutine returns the function $e^{-z}Xi(z)$.

FRESNEL INTEGRAL

The complex Fresnel integral $E(z)$ for complex argument z is defined by the equation

$$E(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \frac{e^t}{t^{\frac{1}{2}}}\, dt$$

where the path of integration lies within that part of the complex plane from which the positive real axis is excluded. The phase of z is in the range 0 to 2π and the phase of $z^{1/2}$ is half the phase of z. The complex Fresnel integral is returned by reference to the call line

CALL FRNLI (M,Z(),F())

If the mode of operation M is 0 then the subroutine returns the function $E(z)$, but if the mode of operation M is 1 then the subroutine returns the function $e^{-z}E(z)$.

ERROR FUNCTION

The complex error function $\operatorname{erf} z$ of argument z is defined by the equation

$$\operatorname{erf} z = \frac{2}{\sqrt{\pi}} \int_{0}^{z} e^{-u^2}\, du = 1 - i\sqrt{2}E(-z^2)$$

The error function is returned by reference to the call line

CALL CERF (M,Z(),F())

If $M = 0$ the subroutine returns $\operatorname{erf} z$, while if $M = 1$ the subroutine returns $1 - \operatorname{erf} z$.

POINT SOURCE

The velocity potential of a point source which moves forward under a free surface is vital in the computation of wave drag on a ship. At a point in the fluid the coordinates are x,y,z where x is forward, y is to the right, and z is downward. The velocity potential φ is the sum of three potentials as expressed by the equation

$$\varphi = \varphi_1 + \varphi_2 + \varphi_3$$

where φ_1 is the potential of the source in an unbounded fluid, φ_2 is the potential of an image source over the free surface, and φ_3 is the potential of the wave train. The source is at depth h below the free surface.

The potential φ_1 is given by the equation

$$\varphi_1 = \frac{1}{\{x^2 + y^2 + (z-h)^2\}^{\frac{1}{2}}}$$

and its derivatives are given by the equations

$$-\frac{\partial\varphi_1}{\partial x} = \frac{x}{\{x^2 + y^2 + (z-h)^2\}^{\frac{3}{2}}}$$

$$-\frac{\partial\varphi_1}{\partial y} = \frac{y}{\{x^2 + y^2 + (z-h)^2\}^{\frac{3}{2}}}$$

$$-\frac{\partial\varphi_1}{\partial z} = \frac{z-h}{\{x^2 + y^2 + (z-h)^2\}^{\frac{3}{2}}}$$

The potential φ_2 is given by the equation

$$\varphi_2 = -\frac{1}{\{x^2 + y^2 + (z+h)^2\}^{\frac{1}{2}}}$$

and its derivatives are given by the equations

$$-\frac{\partial\varphi_2}{\partial x} = -\frac{x}{\{x^2 + y^2 + (z+h)^2\}^{\frac{3}{2}}}$$

$$-\frac{\partial\varphi_2}{\partial y} = -\frac{y}{\{x^2 + y^2 + (z+h)^2\}^{\frac{3}{2}}}$$

$$-\frac{\partial\varphi_2}{\partial z} = -\frac{z+h}{\{x^2 + y^2 + (z+h)^2\}^{\frac{3}{2}}}$$

A parameter κ is defined by the equation

$$\kappa = \frac{g}{U^2}$$

where g is the acceleration of gravity, and U is the speed of the source along the X–axis. A parameter δ is defined by the equation

$$\delta = \frac{\kappa}{\cos^2\theta}\{(z+h) - i(x\cos\theta + y\sin\theta)\}$$

where θ is a variable of integration.

The potential φ_3 is given by the equation

$$\varphi_3 = \frac{\kappa}{\pi} \int_{-\pi}^{+\pi} \frac{e^{-\delta}}{\cos^2\theta} \int_{-\infty}^{\delta} \frac{e^u}{u}\, du\, d\theta$$

and its derivatives are given by the equations

$$-\frac{\partial\varphi_3}{\partial x} = -i\frac{\kappa^2}{\pi} \int_{-\pi}^{+\pi} \frac{e^{-\delta}}{\cos^4\theta} \int_{-\infty}^{\delta} \frac{e^u}{u^2}\, du \cos\theta\, d\theta$$

$$-\frac{\partial\varphi_3}{\partial y} = -i\frac{\kappa^2}{\pi} \int_{-\pi}^{+\pi} \frac{e^{-\delta}}{\cos^4\theta} \int_{-\infty}^{\delta} \frac{e^u}{u^2}\, du \sin\theta\, d\theta$$

$$-\frac{\partial\varphi_3}{\partial z} = +\frac{\kappa^2}{\pi} \int_{-\pi}^{+\pi} \frac{e^{-\delta}}{\cos^4\theta} \int_{-\infty}^{\delta} \frac{e^u}{u^2}\, du\, d\theta$$

The integrand is cyclic but it gets tricky when θ goes to $\frac{1}{2}\pi$. The Fourier representation has many terms. The high accuracy rule of integration for cyclical functions is just the trapezoidal rule, but many panels of integration are still required, and for far coordinates the trapezoidal rule requires a long time. The trapezoidal rule is used by the FORTRAN subroutines CKPSVP and CKPSVF on the main frame. The two subroutines have 184 and 219 lines of program.

An integration by parts makes possible an integration through many cycles of the Fourier terms. The integration by parts is used by the FORTRAN subroutines PSWTVP and PSWTVF on the main frame. The subroutines have 3451 and 3684 lines of program, but they require a constant amount of time for computation. Transfers of these subroutines into BASIC looks overwhelming.

However, it has been possible to translate the integration by trapezoidal rule into classic BASIC.

The potential φ is given by reference to the call line

CALL PSWKVP (K,H,X,Y,Z,P)

where K is parameter κ, H is depth h, and P is potential φ.

The potential gradient is given by reference to the call line

CALL PSWKVF (K,H,X,Y,Z,U,V,W)

where U is velocity forward, V is velocity to the right, and W is velocity downward.

NRL has published a table of velocity potentials for κ=1, h=0.1, z=0 and a few coordinates. Confirmation by subroutine PSWKVP is given in the following table.

x	y	φ
2.00000000	0.00000000	+0.64389683
2.00000000	±1.00000000	+0.62334401
0.00000000	0.00000000	+3.74370189
0.00000000	±1.00000000	+1.54293997
−2.00000000	0.00000000	+2.27604480
−2.00000000	±1.00000000	+5.47025445
−4.00000000	0.00000000	−4.19158598
−4.00000000	±1.00000000	−7.47001522
−6.00000000	0.00000000	+2.22717019
−6.00000000	±1.00000000	+4.26178201
−8.00000000	0.00000000	+2.05501087
−8.00000000	±1.00000000	+2.41167310
−10.00000000	0.00000000	−2.64952399
−10.00000000	±1.00000000	−3.03676767

The tables agree to within the estimate of accuracy by NRL.

LAGRANGE INTERPOLATION

Let $x_1 ... x_N$ be a set of abscissae for which $y_1 ... y_N$ are the ordinates. For each abscissa there is a Lagrange polynomial which is unity at that abscissa and is zero at all other abscissae. An interpolation polynomial which passes through each ordinate is just the sum of the products of ordinates and Lagrange polynomials.

The Lagrange polynomials are initialized by reference to the call line

CALL LGRNX (N,A(),W())

where N is the number of abscissae, A is the address of the abscissae, and W is the address of the polynomials. In the matrix W the element w_{ik} is the coefficient of x^k in the ith Lagrange polynomial. The index i increases downward and the index k increases to the right. An arbitrary function $f(x)$ is given by the summation

$$f(x) = \sum_{i=1}^{N} \sum_{k=0}^{n} f(x_i)\, w_{ik}\, x^k$$

The derivative of $f(x)$ is obtained if x^k is replaced by kx^{k-1} and the integral of $f(x)$ is obtained if x^k is replaced by $x^{k+1}/(k+1)$.

If $n<<N$, then the Lagrange interpolation gives a least squares approximation.

FOURIER TRANSFORM

Let the angle 2π be divided into N equal intervals and let θ_k be the angle at the midpoint of the kth interval. The angle θ_k is given by the equation

$$\theta_k = \frac{2\pi k}{N}$$

where k has the values

$$\tfrac{1}{2} \leq k \leq N-\tfrac{1}{2}$$

Orthogonality is expressed by the equations

$$\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \cos\left(\frac{2\pi mk}{N}\right) \cos\left(\frac{2\pi nk}{N}\right) = 0 \qquad (m \neq n)$$

$$\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \cos\left(\frac{2\pi mk}{N}\right) \sin\left(\frac{2\pi nk}{N}\right) = 0 \qquad \text{by symmetry}$$

$$\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \sin\left(\frac{2\pi mk}{N}\right) \sin\left(\frac{2\pi nk}{N}\right) = 0 \qquad (m \neq n)$$

The orthogonality enables an arbitrary function $f(\theta)$ to be given by the equation

$$f(\theta) = \sum_{m=0}^{M} a_m \cos m\theta + \sum_{m=0}^{M} b_m \sin m\theta$$

where

$$a_m = \frac{\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} f(\theta_k) \cos m\theta_k}{\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \cos^2 m\theta_k} \qquad b_m = \frac{\sum_{k=\frac{1}{2}}^{n-\frac{1}{2}} f(\theta_k) \sin m\theta_k}{\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \sin^2 m\theta_k}$$

The pattern of angles is symmetric when N is a multiple of four.

For any angle the sum of the square of the cosine and the square of the sine is unity. Therefore the denominators satisfy the identity

$$\sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \cos^2 m\theta_k + \sum_{k=\frac{1}{2}}^{N-\frac{1}{2}} \sin^2 m\theta_k \equiv N$$

In the symmetric case the denominators are $0,\frac{1}{2}N,N$ according to the value of m.

The coefficients of the Fourier transform are initialized by reference to the call line

CALL CFOURX (N,M,W())

where N is the number of data, M is the range of multiplicity, and W is the matrix of coefficients. In the matrix k increases downward, and m increases to the right. For each value of m there is a pair of columns in W. The first column of each pair contains $\cos m\theta_k$ while the second column contains $\sin m\theta_k$. Where m=0 the the first column is all 1 and the second column is all 0. Where $m=\frac{1}{2}N$ the first column is all 0 and the second column is ±1.

In the preparation of W only the first four columns require references to COS and SIN. The others are obtained by transfers from the first four.

The trigonometric polynomial is expressed in complex notation by the equations

$$c_m = a_m - ib_m \qquad\qquad f(\theta) = \sum_{m=0}^{M} c_m e^{im\theta}$$

Evaluation of the trigonometric polynomial is obtained by reference to the call line

CALL CFOURV (D,Q,M,C(),F())

where D is the mode of differentiation, Q is the angle argument, M is the number of coefficients, C is the array of coefficients, and F is the complex value.

If the trigonometric polynomial is limited to small order then it is a least squares approximation.

FAST FOURIER TRANSFORM

Let 2π be divided into N equal intervals and let θ_k be defined by the equation

$$\theta_k = \frac{2\pi}{N}k$$

where k has the values

$$0 \leqq k \leqq N-1$$

A function $f(\theta)$ is expressed in terms of the argument θ by the equations

$$f(\theta) = \sum_{m=0}^{N-1} A_m e^{im\theta} \qquad\qquad A_m = \frac{1}{N}\sum_{k=0}^{N-1} f(\theta_k) e^{-km\frac{2\pi}{N}i}$$

If N is 2^n and km is expressed in binary notation then there are many terms in km which cancel the N in the denominator and can be skipped in the evaluation of A_m.

The Fast Fourier Transform is obtained by reference to the call line

CALL CXFFT (M,L,A())

where if the mode M is −1 the transform is from data to coefficients, while if the mode M is +1 the transform is from coefficients to data. The argument L is $\log_2 N$, and the array A holds data or coefficients according to the mode M.

COMPLEX ROOTS

The complex plane is searched for the roots of a given function by subroutine CXRT. Once a root has been located the test function becomes the ratio of the given function to that power polynomial which has the same root. Thus the established root is taken off the list of roots yet to be found. The search for a root proceeds in two stages. The first stage is a hunting stage where a test point takes steps of predetermined length in the direction of steepest descent. The point where the line of steepest descent intercepts the complex plane moves with the test point. The hunting stage terminates when the motion of the intercept point is less than the motion of the test point. The second stage is a homing stage in which the test point goes directly to the intercept point. The homing stage terminates when the intercept point becomes stationary.

The determination of roots is initialized by reference to the call line

```
CALL CXRT (D,E,Z(),N,R())
```

where D is the initial step size in the hunting stage, E is a tolerance for the homing stage, Z is the initial position of the test point, N is the number of roots, and R is the array in which the roots are placed.

The subroutine CXRT makes calls to subroutine FXN for values of the given function and its derivatives. The values are obtained by reference to the call line

```
CALL FXN (M,Z(),F(),D(),S())
```

where M is a mode of operation, Z is the argument, F is the function, D is the first derivative, and S is the second derivative. Modes of computation are listed in the table.

Mode	Computation
M = 0	Function
M = 1	Function and first derivative
M = 2	Function, first and second derivative

At present the subroutine is set up to compute power polynomials, but it can easily be changed to compute any other function.

MATRIX ARITHMETIC

An array is a sequence of elements, where each element is identified by an index. A set of elements may be identified by a two index specification which defines a matrix. The elements in the matrix A are given by the equation

$$A = \left\| \begin{matrix} a_{11} \cdots a_{1n} \\ \cdots \cdots \cdots \\ a_{m1} \cdots a_{mn} \end{matrix} \right\|$$

The dimensions of the matrix are specified by the four-element array

$$l, m, n, k$$

where l = interval between columns
m = number of rows
n = number of columns
k = interval between rows

It takes only a simple loop to lift a matrix out of one array and put it down in another array with a compact specification.

Matrix arithmetic is initialized by reference to the call line

CALL MTRX (M,A(),I(),B(),J(),C(),K())

where M is the mode of operation, A,B,C are the addresses of matrices A,B,C and I,J,K are the specifications for A,B,C. The mode of operation is given in the table

Mode	Operation
0	$B = A$
1	$B = A'$
2	$C = A + B$
3	$C = A - B$
4	$C = A \cdot B$
5	$A \rightarrow A^{-1}$

The matrix is inverted in place and the determinant is stored in the first address of B.

Matrix arithmetic with complex matrices requires only matrix arithmetic with real matrices. Let complex matrices be given by the expressions

$$A+iB \qquad\qquad C+iD$$

Then complex transfer is obtained by the substitutions

$$C \rightarrow A \qquad\qquad D \rightarrow B$$

Complex addition is expressed by the equation

$$(A+iB) + (C+iD) = (A+C) + i(B+D)$$

Complex subtraction is expressed by the equation

$$(A+iB) - (C+iD) = (A-C) + i(B-D)$$

Complex multiplication is expressed by the equation

$$(A+iB) \cdot (C+iD) = (A \cdot C - B \cdot D) + i(A \cdot D + B \cdot C)$$

Complex inversion is expressed by the equation

$$(A+iB)^{-1} = (A + B \cdot A^{-1} \cdot B)^{-1} - i(B + A \cdot B^{-1} \cdot A)^{-1}$$

If matrices and vectors are complex, then the complex expression

$$(A+iB) \cdot (x+iy)$$

is equivalent to the real expression

$$\left\| \begin{matrix} A & -B \\ B & A \end{matrix} \right\| \left\| \begin{matrix} x \\ y \end{matrix} \right\|$$

where the augmented matrix is twice the size of the component matrices. All of these real operations are within the capacity of subroutine MTRX.

The equation $A \cdot x = y$ is solved by the equation $x = A^{-1} \cdot y$. Eliminations are applied to matrix A and to identity I until A is reduced to I and I is reduced to A^{-1}.

Matrix inversion by elimination requires pivot selection. Let a_{ij} be the element in the ith row and the jth column, while a_{rs} is the element in the rth row and the sth column. Elements in the sth column would be eliminated if the rth row were subtracted from the ith row in accordance with the equation

$$a_{ij} = a_{ij} - \frac{a_{is} a_{rj}}{a_{rs}}$$

where a_{rs} is the pivot element. The element a_{ij} would suffer the least rounding error if the elements satisfied the pivot criterion

$$|a_{rs}||a_{ij}| - |a_{is}||a_{rj}| \geqq 0$$

Idependence of the scaling of rows or columns is provided by the criterion

$$\sum_{i=1}^{n} \frac{|a_{rs}||a_{ij}| - |a_{is}||a_{rj}|}{|a_{rs}||a_{ij}| + |a_{is}||a_{rj}|} \geqq 0$$

If j and s are interchanged in the criterion then the sign is reversed. When the criterion is negative the pivot column is moved from s to j. The row number r for the pivot is stored in the sth position of an array R, and the column number s for the pivot is stored in the rth position of an array S.

After pivot selection and elimination the pivot elements are scattered over the intermediate matrix. Let r be the row and let s be the column of a pivot in the intermediate matrix. In the multiplication of the intermediate matrix by the initial matrix let s be at a distance to the right of the diagonal When the rth row of the initial matrix is applied to the rth column of the inverse matrix it must encounter a pivot at the same distance below the diagonal. Then the two pivots interact to put a one on the diagonal. An interchange of rows and an interchange of columns in the intermediate matrix puts the pivot in place below the diagonal The interchange of rows is under the control of R and the interchange of columns is under the control of S.

DETERMINANT

A set of equations

$$a_{11}x_1 + \cdots + a_{1n}x_n = y_1$$
$$\cdots$$
$$a_{n1}x_1 + \cdots + a_{nn}x_n = y_n$$

has an $n \times n$ matrix A which is expressed by the equation

$$A = \begin{Vmatrix} a_{11} \cdots a_{1n} \\ \cdots \; \cdots \; \cdots \\ a_{n1} \cdots a_{nn} \end{Vmatrix}$$

and has a determinant |A| which is expressed by the equation

$$|A| = \begin{vmatrix} a_{11} \cdots a_{1n} \\ \cdots \; \cdots \; \cdots \\ a_{n1} \cdots a_{nn} \end{vmatrix}$$

When an element a_{ij} is in a compact array, the index number in the array is $in-n+j$.

The solution of the system of equations is given by Cramer's Rule:

$$x_1 = \frac{\begin{vmatrix} y_1 \cdots a_{1n} \\ \cdots \; \cdots \; \cdots \\ y_n \cdots a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} \cdots a_{1n} \\ \cdots \; \cdots \; \cdots \\ a_{n1} \cdots a_{nn} \end{vmatrix}} \qquad x_n = \frac{\begin{vmatrix} a_{11} \cdots y_1 \\ \cdots \; \cdots \; \cdots \\ a_{n1} \cdots y_n \end{vmatrix}}{\begin{vmatrix} a_{11} \cdots a_{1n} \\ \cdots \; \cdots \; \cdots \\ a_{n1} \cdots a_{nn} \end{vmatrix}}$$

Verification of the solution is possible when $y_1 \cdots y_n$ are replaced by their expressions in terms of $x_1 \cdots x_n$.

By definition the determinant of A is the sum of terms of which the first term is the product of the diagonal elements and the additional terms are the products of elements with permuted indices and with sign reversal for each permutation of indices.

If any row or any column of a determinant is multiplied throughout by the same factor, then the determinant is multiplied by that factor. If any row or any column of a determinant is the sum of two rows or two columns, then the determinant splits into the sum of two determinants with each row or each column. If a determinant has any two rows or any two columns identical, then the determinant is zero. If two rows or two columns are interchanged the determinant is reversed in sign. If a row or a column is multiplied throughout by the same factor and is added to another row or column, the determinant is not changed.

The determinant |A−QI| is evaluated by reference to the call line

```
CALL DTMX (Q,A(),I(),D)
```

where Q is an argument, A is the address of matrix A, I is the specification of matrix A, and D is the determinant of matrix A−QI.

CHARACTERISTIC

If the components $v_1 \cdots v_n$ of the vector v satisfy the homogeneous equations

$$a_{11}v_1 + \cdots + a_{1n}v_n = \alpha v_1$$

$$\cdots$$

$$a_{n1}v_1 + \cdots + a_{nn}v_n = \alpha v_n$$

then α is a characteristic root and v is a characteristic vector of the matrix A.

According to Cramer's rule the determinant $|A-\alpha I|$ must be zero. The determinant is a power polynomial of the nth degree in α. The roots of the polynomial are the characteristic roots of the matrix A.

The characteristic vector for each characteristic root is found by pivot selection and elimination in the matrix $A-\alpha I$. Elements in the rth column are eliminated when the rth row is subtracted from the ith row as in the equation

$$a_{ij} = a_{ij} - \frac{a_{ir}a_{rj}}{a_{rr}}$$

where a_{rr} is the pivot element. The element a_{ij} suffers the least rounding error if the elements satisfy the scale invariant criterion

$$\sum_{j=r}^{n} \frac{|a_{rr}||a_{ij}| - |a_{ir}||a_{rj}|}{|a_{rr}||a_{ij}| + |a_{ir}||a_{rj}|} \geqq 0$$

If rows i and r are interchanged then the sign of the criterion is reversed. The rows i and r are interchanged if the criterion is negative.

After pivot selection and elimination what is left of the matrix is an upper right triangle. The determinant of the triangle is the product of the diagonal elements. Inasmuch as the determinant of a characteristic matrix is zero, there must be at least one diagonal element equal to zero. The component of vector in the same column with the zero element is arbitrary. The components on the right of the arbitrary component are zero, while the components on the left of the arbitrary component are determined by back substitution. If there is more than one zero on the diagonal, and if the element in the same row as the upper zero and in the same column as the lower zero is also zero, then there are two vectors for the same root. Otherwise the matrix is defective.

It has been necessary to introduce a tolerance ϵ such that any number less than the tolerance is interpreted as a zero with noise, while any number greater than the tolerance is interpreted as a nonzero with noise.

The characteristic vector for a particular root is obtained by reference to the call line

```
CALL RVMX (E,M,R,A(),I(),V())
```

where E is the tolerance ϵ, M is the mode of operation, R is the address of the root, A is the address of the matrix, I is the specification of the matrix, and V is the address of the vector. M is 0 for initialization, and M is 1 for continuation.

DISCUSSION

The rational approximations for special functions were pushed to sixteen digit accuracy, but existing computers have only fourteen digit accuracy unless they have a coprocessor.

If an iteration is applied to an entire matrix, there can be a problem of slow convergence, but if an elimination of elements is applied to the matrix, there can be a problem of rounding error.

A chain of operations with pivot selection is optimum. A set of values of the characteristic parameter can be presented to subroutine DTMX to obtain a set of values for the characteristic determinant. The set of values of the characteristic determinant is presented to the subroutine LGRNX to obtain the coefficients of the characteristic polynomial. The characteristic polynomial is presented to the subroutine CXRT to obtain the characteristic roots. The characteristic roots are presented to the subroutine RVMX to obtain the characteristic vectors. This procedure was successful with every 4×4 matrix which could be found in Bureau of Standards publications.

An interesting matrix is the permutation

$$A = \begin{Vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{Vmatrix}$$

Multiplication of any vector by the matrix reverses the order of the components of the vector. Any vector with components which are symmetric with respect to the center is a characteristic vector with characteristic root +1. Any vector with components which are antisymmetric with respect to the center, is a characteristic vector with characteristic root −1. The product of the matrix by itself is the identity matrix. The matrix is its own inverse, and it is a square root of the identity matrix.

All of the subroutines in the showcase were numbered by hand and were keyed in by hand. Not all of the subroutines in the showcase were translated from FORTRAN into BASIC. Not included are two files PSWTVP and PSWTVF on the main frame. They use integration by parts to compute the velocity potential and the potential gradient of a point source under a free surface. The two files contain twelve subroutines each.

The files were prepared when we still had punched cards. Each subroutine was in a deck of cards. It was easy to stack the decks in a tray and hand the tray to the operator.

Now we are no longer allowed to use punched cards. The two files of subroutines can still be used on the main frame if a main program is appended to either file. An essential instruction in the main program is

```
//NAME EXEC VSF2CLG
```

where CLG stands for compile, link, go. The FORTRAN compiler takes control and carries execution to completion. Appendage of the main program is accomplished with the aid of the utility XEDIT which has GET and PUT.

Concatenation is not so easy in True BASIC. If the WINDOWS menu is pulled down to COMMAND, then a command window appears on the screen. Instructions in the command window modify a file which is on display in the screen.

With the first of two files on display in the screen the instruction

```
INCLUDE NAME
```

will attach the file with the name NAME to the end of the first file. The instruction

```
DO RENUM, I,K
```

renumbers the lines with the initial value I and the increment K. The line numbers must be in sequence before they can be renumbered.

Another method of concatenation requires a departure of each subroutine from standard format. The first line of each subroutine is replaced with the statement

```
1 EXTERNAL
```

The SUB statement is moved down into the space between asterisks. A main program has a stack of statements

```
LIBRARY "NAME"
```

where NAME is the name of each subroutine. This has the advantage that the subroutines do not have to be renumbered. It is not clear why the computer cannot discover for itself that SUB is followed by an external program.

Maybe some day all this will be attempted, but today is not the day.

CONCLUSION

Experiments with the subroutine showcase have been instructive and even *fun.*

BIBLIOGRAPHY

1 *Modern Analysis*
E.T.Whittaker and G. N. Watson (Combridge University Press, 1952)

2 *The Theory of Bessel Functions*
G. N. Watson, (Cambridge University Press, 1962)

3 *Handbook of Mathematical Functions*
M. Abramowitz and I. A. Stegun (Dover Publications, Inc. New York, 1972)

4 *Simultaneous Linear Equations and the Determination of Eigenvalues*
L. J. Paige and O. Taussky, National Bureau of Standards Applied Mathematics Series #29 (Aug 1953)

5 *Simultaneous Linear Equations and the Determination of Eigenvalues*
E. L. Stiefel, P. Henrici, and H. Rutishauser, National Bureau of Standards Applied Mathematics Series #49 (Jan 1958)

6 *Basic Theorems in Matrix Theory*
M. Marcus, National Bureau of Standards Applied Mathematics Series #57 (Jan 1960)

7 *Polynomial Arithmetic and Contour Construction*
A. V. Hershey, NSWC/DL Report TR-3688 (Oct 1977)

8 *Matrix Arithmetic and Characteristics Computation*
A, V. Hershey, NSWC/DL Report TR-3689 (Oct 1977)

9 *NSWC Library of Mathematics Subroutines*
A. H. Morris, Jr. NSWC Report NSWCDD/TR-92/425 (Jan 1993)

10 *New Formulas for Computing Incomplete Elliptic Integrals of the First and Second Kind*
A. R. DiDonato and A. V. Hershey, Journal of the Association for Computing Machinery 6 515 (1959)

11 *Approximations of Functions by Sets of Poles*
A. V. Hershey, NWL Technical Report TR-2564 (July 1971)

12 *Computation of Special Functions*
A. V. Hershey, NSWC/DL Report TR-3788 (Nov 1978)

13 *FORTRAN Programming for Surface Wave Trains*
A. V. Hershey, NWL Technical Report TR-2714 (Sept 1972)

14 *Computation of Velocity in the Wave Train of a Point Source*
A. V. Hershey, NSWC/DL Report TR-3720 (Sept 1977)

15 *Calculation of the Odd and Even Integral Components of the Wave Resistance Green's Function*
H. T. Wang and J. C. W. Rogers, NRL Memorandum Report 6411 (Sept 1989)

16 *Cartography and Typography with True BASIC*
A. V. Hershey, NPS Report NPS-09-95-003 (Sept 1995)

APPENDIX A

PROGRAMS

```
1 SUB EXPN (X,Y)
2 !**********************
3 !EXPONENTIAL FUNCTION
4 !**********************
5 OPTION NOLET
6 W=0
7 IF (X<-709) THEN 25
8 E=2.718281828459045
9 U=ABS(X)
10 W=1
11 IF (U<1) THEN 15
12 W=E*W
13 U=U-1
14 GOTO 11
15 S=0
16 T=1
17 N=0
18 S=S+T
19 N=N+1
20 T=T*U/N
21 IF (S+T>S) THEN 18
22 W=S*W
23 IF (X>=0) THEN 25
24 W=1/W
25 Y=W
26 END SUB
27 OPTION NOLET
28 X=1.9459101490553133
29 CALL EXPN (X,Y)
30 PRINT USING " +#.###############^^^^^": X,Y,EXP(X)
31 END
```

EXPN

```
1 SUB LOGM (X,Y)
2 !************************
3 !LOGARITHMIC FUNCTION
4 !************************
5 OPTION NOLET
6 Q=6.9314718055994531e-01
7 R=1.4142135623730950e+00
8 L=0
9 U=X
10 IF (U<=0) THEN STOP
11 IF (U<1) THEN 16
12 IF (U<R) THEN 20
13 U=U/2
14 L=L+Q
15 GOTO 12
16 IF (U>1/R) THEN 20
17 U=2*U
18 L=L-Q
19 GOTO 16
20 U=U-1
21 S=0
22 T=U
23 N=1
24 S=S+T/N
25 N=N+1
26 T=-U*T
27 IF (ABS(S)+ABS(T/N)>ABS(S)) THEN 24
28 L=L+S
29 Y=L
30 END SUB
31 OPTION NOLET
32 X=7
33 CALL LOGM (X,Y)
34 PRINT USING " +#.###############^^^^": X,Y
35 END
```

LOGM

```
1 SUB ELLPCI (A, K, F, E)
2 !**********************************
3 !SUBROUTINE FOR ELLIPTIC INTEGRALS
4 !**********************************
5 ! A = ARC = phi
6 ! K = MODULUS = K
7 ! F = INTEGRAL OF FIRST KIND
8 ! E = INTEGRAL OF SECOND KIND
9 !
10 OPTION NOLET
11 P2=K*K
12 Q2=(1-K)*(1+K)
13 S=SIN(A)
14 C=COS(A)
15 S0=S*S
16 C0=C*C
17 P0=ABS(K*S)
18 Q0=ABS(K*C)
19 IF (P0>=SQR(.5)) THEN 44
20 R1=A
21 R2=1
22 S1=0
23 S2=0
24 N1=1
25 N2=2
26 T0=A*S0
27 T7=S*C
28 GOTO 33
29 N1=N2+1
30 N2=N1+1
31 T0=S0*T0
32 T7=S0*T7
33 R0=S1
34 R1=(N1*R1-T7)/N2
35 R2=P2*R2/N2
36 S2=S2+R1*R2
37 R2=N1*R2
38 S1=S1+R1*R2
39 IF (ABS(T0)<ABS(R0)) THEN 41
40 IF (ABS(S1)>ABS(R0)) THEN 29
41 F=A+S1
42 E=A-S2
43 EXIT SUB
44 D7=(1-P0)*(1+P0)
45 D0=SQR(D7)
46 I2=1
47 J2=1
48 K2=0
49 M7=0
50 N7=0
```

ELLPCI

```
51 S1=0
52 S2=0
53 S3=0
54 S4=0
55 T3=Q0*D0
56 N0=2
57 GOTO 63
58 I2=I1
59 J2=J1
60 K2=K1
61 N0=N0+2
62 T3=D7*T3
63 N1=(N0-1)/N0
64 N2=(N0+1)/(N0+2)
65 I1=N1*I2
66 J1=N1*N1*Q2*J2
67 K1=K2+2/(N0*(N0-1))
68 R0=T3/N0
69 M7=N2*N2*Q2*(M7-R0*I1)
70 N7=N1*N2*Q2*(N7-R0*I2)
71 D1=J1
72 D2=N2*J1
73 D3=M7-J1*K1
74 D4=N7-N1*Q2*J2*K1+Q2*J2/(N0*N0)
75 R0=S3
76 S1=S1+D1
77 S2=S2+D2
78 S3=S3+D3
79 S4=S4+D4
80 IF (S3<R0) THEN 58
81 T0=D0+Q0
82 R7=T0/4
83 L1=-LOG(R7)
84 T7=1+P0
85 S7=T7/2
86 L2=+LOG(S7)
87 T1=(1+S1)*L1+Q0/D0*L2
88 T2=(.5+S2)*Q2*L1+1-Q0/D0*(1-P0)
89 F=T1+S3
90 E=T2+S4
91 IF (A>=0) THEN 94
92 F=-F
93 E=-E
94 EXIT SUB
95 END SUB
96 OPTION NOLET
97 Q=30/180*PI
98 K=SIN(Q)
99 A=30/180*PI
100 CALL ELLPCI (A, K, F, E)
```

ELLPCI

```
101 PRINT USING "+#.##########^^^^": F
102 PRINT USING "+#.##########^^^^": E
103 END
```

```
1 SUB BSSLJ (Z(),N,J())
2 !********************************
3 !BASIC SUBROUTINE FOR BESSEL J
4 !********************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (INTEGER)
7 !J = FUNCTION (COMPLEX ARRAY)
8 !
9 OPTION NOLET
10 IF (H=1) THEN 88
11 H=1
12 DIM D0(15),D1(15),E0(15),E1(15)
13 DIM R0(2),Z0(2),R1(2),S1(2),T1(2)
14 DIM R7(4),S7(4),T7(2),Q7(2),F1(2)
15 DATA  0.000000000000000E00
16 DATA -1.648995051422117E-2
17 DATA -7.186218800685365E-2
18 DATA -1.670868781248656E-1
19 DATA -3.025822502194688E-1
20 DATA -4.806139452459267E-1
21 DATA -7.070752393578979E-1
22 DATA -9.929957905395160E-1
23 DATA -1.355839256125922E00
24 DATA -1.821059078991320E00
25 DATA -2.424821753108787E00
26 DATA -3.219566557087496E00
27 DATA -4.286580772483836E00
28 DATA -5.770228167981279E00
29 DATA -8.013712609525260E00
30 FOR K=1 TO 15
31 READ D0(K)
32 NEXT K
33 DATA  0.000000000000000E00
34 DATA -5.577424298795054E-3
35 DATA -4.991129441724757E-2
36 DATA -1.374409116523968E-1
37 DATA -2.672337847105657E-1
38 DATA -4.403801668086817E-1
39 DATA -6.618136148725406E-1
40 DATA -9.418610776650166E-1
41 DATA -1.297541304683261E00
42 DATA -1.754076967198164E00
43 DATA -2.347552998822763E00
44 DATA -3.130413326891964E00
45 DATA -4.183971205637291E00
46 DATA -5.652517992149936E00
47 DATA -7.878639598106769E00
48 FOR K=1 TO 15
49 READ D1(K)
50 NEXT K
```

```
51 DATA  0.000000000000000E00
52 DATA -4.809423363874473E-3
53 DATA -1.313662003477595E-2
54 DATA -1.948438340084579E-2
55 DATA -2.199489000320033E-2
56 DATA -2.093966256765194E-2
57 DATA -1.746002684586503E-2
58 DATA -1.279378133620848E-2
59 DATA -8.052344217965918E-3
60 DATA -4.158173750027601E-3
61 DATA -1.643177387479224E-3
62 DATA -4.491755853147087E-4
63 DATA -7.285947655740069E-5
64 DATA -5.382652306582855E-6
65 DATA -9.937790480362892E-8
66 FOR K=1 TO 15
67 READ E0(K)
68 NEXT K
69 DATA  0.000000000000000E00
70 DATA  7.538057792005914E-2
71 DATA  7.122935374034643E-2
72 DATA  6.331162242281997E-2
73 DATA  5.282402645233011E-2
74 DATA  4.133053594414916E-2
75 DATA  3.013505739475096E-2
76 DATA  2.010434395927201E-2
77 DATA  1.185522230680744E-2
78 DATA  5.860555109560099E-3
79 DATA  2.254651482673253E-3
80 DATA  6.081730415363355E-4
81 DATA  9.842155506257467E-5
82 DATA  7.321390930380890E-6
83 DATA  1.372796673846658E-7
84 FOR K=1 TO 15
85 READ E1(K)
86 NEXT K
87 RESTORE
88 Z2=Z(1)*Z(1)+Z(2)*Z(2)
89 Z1=SQR(Z2)
90 N0=ABS(N)
91 S0=+1
92 IF (N>=0) THEN 95
93 IF (N=2*IP(N/2)) THEN 95
94 S0=-1
95 IF (Z(1)>=0) THEN 101
96 Z0(1)=-Z(1)
97 Z0(2)=-Z(2)
98 IF (N=2*IP(N/2)) THEN 103
99 S0=-S0
100 GOTO 103
```

```
101 Z0(1)=+Z(1)
102 Z0(2)=+Z(2)
103 IF (Z1<=17.5+.5*N0*N0) THEN 106
104 N1=N0
105 GOTO 122
106 N1=.5*Z1-.5*ABS(Z0(2))+.5*ABS(.5*Z1-ABS(Z0(2)))
107 IF (N0<=N1) THEN 112
108 N1=+IP(.25*Z2)
109 IF (N0<=N1) THEN 283
110 N1=N0
111 GOTO 283
112 IF (Z1<=17.5) THEN 115
113 N1=+IP(SQR(2*(Z1-17.5)))
114 GOTO 122
115 IF (Z2<1) THEN 117
116 IF (-ABS(Z(2))+.096*Z(1)*Z(1)>=0) THEN 121
117 N1=+IP(.25*Z2)
118 IF (N0<=N1) THEN 283
119 N1=N0
120 GOTO 283
121 N1=0
122 S1(1)=Z0(1)
123 S1(2)=Z0(2)
124 M1=S0*SQR(2/PI)
125 R0(1)=SQR(S1(1)+Z1)
126 R0(2)=S1(2)/R0(1)
127 R0(1)=SQR(.5)*R0(1)
128 R0(2)=SQR(.5)*R0(2)
129 F1(1)=+M1*R0(1)/Z1
130 F1(2)=-M1*R0(2)/Z1
131 IF (Z1<=17.5) THEN 163
132 R1(1)=+.5*Z0(1)/Z2
133 R1(2)=-.5*Z0(2)/Z2
134 N2=N1*N1-.25
135 S7(1)=0
136 S7(2)=0
137 S7(3)=0
138 S7(4)=0
139 T7(1)=1
140 T7(2)=0
141 M0=0
142 GOTO 149
143 N2=N2-2*M0
144 M0=M0+1
145 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
146 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
147 T7(1)=-N2*T1(1)/M0
148 T7(2)=-N2*T1(2)/M0
149 S7(1)=S7(1)+T7(1)
150 S7(2)=S7(2)+T7(2)
```

```
151 N2=N2-2*M0
152 M0=M0+1
153 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
154 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
155 T7(1)=+N2*T1(1)/M0
156 T7(2)=+N2*T1(2)/M0
157 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 159
158 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 189
159 S7(3)=S7(3)+T7(1)
160 S7(4)=S7(4)+T7(2)
161 IF (M0<35) THEN 143
162 GOTO 189
163 S7(1)=1
164 S7(2)=0
165 S7(3)=1
166 S7(4)=0
167 FOR K=2 TO 15
168 T1(1)=+Z0(2)-D0(K)
169 T1(2)=-Z0(1)
170 S2=T1(1)*T1(1)+T1(2)*T1(2)
171 T7(1)=+E0(K)*T1(1)/S2
172 T7(2)=-E0(K)*T1(2)/S2
173 S7(1)=S7(1)+T7(1)
174 S7(2)=S7(2)+T7(2)
175 T1(1)=-Z0(2)-D0(K)
176 T1(2)=+Z0(1)
177 S2=T1(1)*T1(1)+T1(2)*T1(2)
178 T7(1)=+E0(K)*T1(1)/S2
179 T7(2)=-E0(K)*T1(2)/S2
180 S7(3)=S7(3)+T7(1)
181 S7(4)=S7(4)+T7(2)
182 NEXT K
183 T1(1)=+.5*(S7(2)-S7(4))
184 T1(2)=-.5*(S7(1)-S7(3))
185 S7(1)=+.5*(S7(1)+S7(3))
186 S7(2)=+.5*(S7(2)+S7(4))
187 S7(3)=T1(1)
188 S7(4)=T1(2)
189 Q7(1)=Z0(1)-.5*PI*(N1+.5)
190 Q7(2)=Z0(2)
191 T1(1)=+COS(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
192 T1(2)=-SIN(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
193 T7(1)=S7(1)*T1(1)-S7(2)*T1(2)
194 T7(2)=S7(1)*T1(2)+S7(2)*T1(1)
195 T1(1)=+SIN(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
196 T1(2)=+COS(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
197 R7(1)=T7(1)-S7(3)*T1(1)+S7(4)*T1(2)
198 R7(2)=T7(2)-S7(3)*T1(2)-S7(4)*T1(1)
199 IF (N1=N0) THEN 280
200 R7(3)=R7(1)
```

```
201 R7(4)=R7(2)
202 N1=N1+1
203 IF (Z1<=17.5) THEN 233
204 N2=N1*N1-.25
205 S7(1)=0
206 S7(2)=0
207 S7(3)=0
208 S7(4)=0
209 T7(1)=1
210 T7(2)=0
211 M0=0
212 GOTO 219
213 N2=N2-2*M0
214 M0=M0+1
215 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
216 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
217 T7(1)=-N2*T1(1)/M0
218 T7(2)=-N2*T1(2)/M0
219 S7(1)=S7(1)+T7(1)
220 S7(2)=S7(2)+T7(2)
221 N2=N2-2*M0
222 M0=M0+1
223 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
224 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
225 T7(1)=+N2*T1(1)/M0
226 T7(2)=+N2*T1(2)/M0
227 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 229
228 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 259
229 S7(3)=S7(3)+T7(1)
230 S7(4)=S7(4)+T7(2)
231 IF (M0<35) THEN 213
232 GOTO 259
233 S7(1)=1
234 S7(2)=0
235 S7(3)=1
236 S7(4)=0
237 FOR K=2 TO 15
238 T1(1)=+Z0(2)-D1(K)
239 T1(2)=-Z0(1)
240 S2=T1(1)*T1(1)+T1(2)*T1(2)
241 T7(1)=+E1(K)*T1(1)/S2
242 T7(2)=-E1(K)*T1(2)/S2
243 S7(1)=S7(1)+T7(1)
244 S7(2)=S7(2)+T7(2)
245 T1(1)=-Z0(2)-D1(K)
246 T1(2)=+Z0(1)
247 S2=T1(1)*T1(1)+T1(2)*T1(2)
248 T7(1)=+E1(K)*T1(1)/S2
249 T7(2)=-E1(K)*T1(2)/S2
250 S7(3)=S7(3)+T7(1)
```

```
251 S7(4)=S7(4)+T7(2)
252 NEXT K
253 T1(1)=+.5*(S7(2)-S7(4))
254 T1(2)=-.5*(S7(1)-S7(3))
255 S7(1)=+.5*(S7(1)+S7(3))
256 S7(2)=+.5*(S7(2)+S7(4))
257 S7(3)=T1(1)
258 S7(4)=T1(2)
259 Q7(1)=Z0(1)-.5*PI*(N1+.5)
260 Q7(2)=Z0(2)
261 T1(1)=+COS(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
262 T1(2)=-SIN(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
263 T7(1)=S7(1)*T1(1)-S7(2)*T1(2)
264 T7(2)=S7(1)*T1(2)+S7(2)*T1(1)
265 T1(1)=+SIN(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
266 T1(2)=+COS(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
267 R7(1)=T7(1)-S7(3)*T1(1)+S7(4)*T1(2)
268 R7(2)=T7(2)-S7(3)*T1(2)-S7(4)*T1(1)
269 GOTO 279
270 T7(1)=+2*N1*Z0(1)/Z2
271 T7(2)=-2*N1*Z0(2)/Z2
272 T1(1)=T7(1)*R7(1)-T7(2)*R7(2)-R7(3)
273 T1(2)=T7(1)*R7(2)+T7(2)*R7(1)-R7(4)
274 R7(3)=R7(1)
275 R7(4)=R7(2)
276 R7(1)=T1(1)
277 R7(2)=T1(2)
278 N1=N1+1
279 IF (N1<N0) THEN 270
280 J(1)=F1(1)*R7(1)-F1(2)*R7(2)
281 J(2)=F1(1)*R7(2)+F1(2)*R7(1)
282 EXIT SUB
283 S1(1)=+.25*(Z0(1)*Z0(1)-Z0(2)*Z0(2))
284 S1(2)=+.5*Z0(1)*Z0(2)
285 N2=N1
286 S7(1)=0
287 S7(2)=0
288 S7(3)=0
289 S7(4)=0
290 T7(1)=1
291 T7(2)=0
292 M0=0
293 N2=N2+1
294 T1(1)=+T7(1)/N2
295 T1(2)=+T7(2)/N2
296 S7(3)=S7(3)+T1(1)
297 S7(4)=S7(4)+T1(2)
298 T7(1)=-T1(1)*S1(1)+T1(2)*S1(2)
299 T7(2)=-T1(1)*S1(2)-T1(2)*S1(1)
300 M0=M0+1
```

```
301 T7(1)=T7(1)/M0
302 T7(2)=T7(2)/M0
303 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 305
304 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 308
305 S7(1)=S7(1)+T7(1)
306 S7(2)=S7(2)+T7(2)
307 GOTO 293
308 S7(1)=S7(1)+1
309 N2=N1+1
310 S7(3)=N2*S7(3)
311 S7(4)=N2*S7(4)
312 GOTO 323
313 N2=N1*(N1+1)
314 T7(1)=S1(1)/N2
315 T7(2)=S1(2)/N2
316 T1(1)=-T7(1)*S7(3)+T7(2)*S7(4)
317 T1(2)=-T7(1)*S7(4)-T7(2)*S7(3)
318 S7(3)=S7(1)
319 S7(4)=S7(2)
320 S7(1)=S7(1)+T1(1)
321 S7(2)=S7(2)+T1(2)
322 N1=N1-1
323 IF (N1>N0) THEN 313
324 F1(1)=S0
325 F1(2)=0
326 N1=0
327 GOTO 333
328 N1=N1+1
329 T7(1)=F1(1)*Z0(1)-F1(2)*Z0(2)
330 T7(2)=F1(1)*Z0(2)+F1(2)*Z0(1)
331 F1(1)=.5*T7(1)/N1
332 F1(2)=.5*T7(2)/N1
333 IF (N1<N0) THEN 328
334 J(1)=F1(1)*S7(1)-F1(2)*S7(2)
335 J(2)=F1(1)*S7(2)+F1(2)*S7(1)
336 EXIT SUB
337 END SUB
338 OPTION NOLET
339 DIM Z(2),J(2)
340 Z(1)=7
341 Z(2)=0
342 N=0
343 CALL BSSLJ (Z(),N,J())
344 PRINT USING "+#.##########^^^^": J(1)
345 PRINT USING "+#.##########^^^^": J(2)
346 N=1
347 CALL BSSLJ (Z(),N,J())
348 PRINT USING "+#.##########^^^^": J(1)
349 PRINT USING "+#.##########^^^^": J(2)
350 END
```

```
1 SUB BSSLY (Z(),N,Y())
2 !********************************
3 !BASIC SUBROUTINE FOR BESSEL Y
4 !********************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (INTEGER)
7 !Y = FUNCTION (COMPLEX ARRAY)
8 !
9 OPTION NOLET
10 IF (H=1) THEN 89
11 H=1
12 DIM D(30),E(30),Z0(2),R1(2),S1(2),L1(2),T1(2)
13 DIM S7(4),T7(4),L7(2),Q2(2),R2(2),Q7(2),F1(2)
14 DATA 0.5772156649015329E00
15 READ G
16 DATA  0.000000000000000E00
17 DATA -1.648995051422117E-2
18 DATA -7.186218800685365E-2
19 DATA -1.670868781248656E-1
20 DATA -3.025822502194688E-1
21 DATA -4.806139452459267E-1
22 DATA -7.070752393578979E-1
23 DATA -9.929957905395160E-1
24 DATA -1.355839256125922E00
25 DATA -1.821059078991320E00
26 DATA -2.424821753108787E00
27 DATA -3.219566557087496E00
28 DATA -4.286580772483836E00
29 DATA -5.770228167981279E00
30 DATA -8.013712609525260E00
31 FOR I=1 TO 15
32 READ D(I)
33 NEXT I
34 DATA  0.000000000000000E00
35 DATA -5.577424298795054E-3
36 DATA -4.991129441724757E-2
37 DATA -1.374409116523968E-1
38 DATA -2.672337847105657E-1
39 DATA -4.403801668086817E-1
40 DATA -6.618136148725406E-1
41 DATA -9.418610776650166E-1
42 DATA -1.297541304683261E00
43 DATA -1.754076967198164E00
44 DATA -2.347552998822763E00
45 DATA -3.130413326891964E00
46 DATA -4.183971205637291E00
47 DATA -5.652517992149936E00
48 DATA -7.878639598106769E00
49 FOR I=16 TO 30
50 READ D(I)
```

```
 51 NEXT I
 52 DATA  0.000000000000000E00
 53 DATA -4.809423363874473E-3
 54 DATA -1.313662003477595E-2
 55 DATA -1.948438340084579E-2
 56 DATA -2.199489000320033E-2
 57 DATA -2.093966256765194E-2
 58 DATA -1.746002684586503E-2
 59 DATA -1.279378133620848E-2
 60 DATA -8.052344217965918E-3
 61 DATA -4.158173750027601E-3
 62 DATA -1.643177387479224E-3
 63 DATA -4.491755853147087E-4
 64 DATA -7.285947655740069E-5
 65 DATA -5.382652306582855E-6
 66 DATA -9.937790480362892E-8
 67 FOR I=1 TO 15
 68 READ E(I)
 69 NEXT I
 70 DATA  0.000000000000000E00
 71 DATA  7.538057792005914E-2
 72 DATA  7.122935374034643E-2
 73 DATA  6.331162242281997E-2
 74 DATA  5.282402645233011E-2
 75 DATA  4.133053594414916E-2
 76 DATA  3.013505739475096E-2
 77 DATA  2.010434395927201E-2
 78 DATA  1.185522230680744E-2
 79 DATA  5.860555109560099E-3
 80 DATA  2.254651482673253E-3
 81 DATA  6.081730415363355E-4
 82 DATA  9.842155506257467E-5
 83 DATA  7.321390930380890E-6
 84 DATA  1.372796673846658E-7
 85 FOR I=16 TO 30
 86 READ E(I)
 87 NEXT I
 88 RESTORE
 89 Z2=Z(1)*Z(1)+Z(2)*Z(2)
 90 L1(1)=.5*LOG(Z2)
 91 X1=Z(1)
 92 Y1=Z(2)
 93 GOSUB 345
 94 L1(2)=Q1
 95 N0=ABS(N)
 96 S0=+1
 97 IF (N>0) THEN 100
 98 IF (N=2*IP(N/2)) THEN 100
 99 S0=-1
100 IF (Z(1)>0) THEN 104
```

```
101 Z0(1)=-Z(1)
102 Z0(2)=-Z(2)
103 GOTO 106
104 Z0(1)=+Z(1)
105 Z0(2)=+Z(2)
106 IF (Z2<=1) THEN 149
107 IF (Z2>=289) THEN 109
108 IF (-ABS(Z(2))+.096*Z(1)*Z(1)<=0) THEN 149 ELSE 129
109 M1=S0*SQR(2/PI)*EXP(-.5*L1(1))
110 F1(1)=M1*COS(-.5*L1(2))
111 F1(2)=M1*SIN(-.5*L1(2))
112 IF (N0>1) THEN 120
113 N7=N0
114 GOSUB 182
115 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
116 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
117 S7(1)=T1(1)
118 S7(2)=T1(2)
119 GOTO 179
120 N7=1
121 GOSUB 182
122 Q2(1)=-F1(1)*S7(1)+F1(2)*S7(2)
123 Q2(2)=-F1(1)*S7(2)-F1(2)*S7(1)
124 N7=0
125 GOSUB 182
126 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
127 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
128 GOTO 167
129 M1=S0/SQR(2*PI)*EXP(-.5*L1(1))
130 F1(1)=M1*COS(-.5*L1(2))
131 F1(2)=M1*SIN(-.5*L1(2))
132 IF (N0>1) THEN 140
133 N7=N0
134 GOSUB 240
135 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
136 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
137 S7(1)=T1(1)
138 S7(2)=T1(2)
139 GOTO 179
140 N7=1
141 GOSUB 240
142 Q2(1)=-F1(1)*S7(1)+F1(2)*S7(2)
143 Q2(2)=-F1(1)*S7(2)-F1(2)*S7(1)
144 N7=0
145 GOSUB 240
146 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
147 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
148 GOTO 167
149 F1(1)=S0*(2/PI)
150 F1(2)=0
```

BSSLY

```
151 IF (N0>1) THEN 159
152 N7=N0
153 GOSUB 290
154 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
155 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
156 S7(1)=T1(1)
157 S7(2)=T1(2)
158 GOTO 179
159 N7=1
160 GOSUB 290
161 Q2(1)=-F1(1)*S7(1)+F1(2)*S7(2)
162 Q2(2)=-F1(1)*S7(2)-F1(2)*S7(1)
163 N7=0
164 GOSUB 290
165 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
166 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
167 R1(1)=+Z(1)/Z2
168 R1(2)=-Z(2)/Z2
169 N7=0
170 GOTO 175
171 Q2(1)=R2(1)
172 Q2(2)=R2(2)
173 R2(1)=S7(1)
174 R2(2)=S7(2)
175 S7(1)=2*N7*(R1(1)*R2(1)-R1(2)*R2(2))-Q2(1)
176 S7(2)=2*N7*(R1(1)*R2(2)+R1(2)*R2(1))-Q2(2)
177 N7=N7+1
178 IF (N7<N0) THEN 171
179 Y(1)=S7(1)
180 Y(2)=S7(2)
181 EXIT SUB
182 S7(1)=0
183 S7(2)=0
184 S7(3)=0
185 S7(4)=0
186 R1(1)=+.5*Z0(1)/Z2
187 R1(2)=-.5*Z0(2)/Z2
188 N1=N7*N7-.25
189 T7(1)=1
190 T7(2)=0
191 M7=0
192 GOTO 199
193 N1=N1-2*M7
194 M7=M7+1
195 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
196 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
197 T7(1)=-N1*T1(1)/M7
198 T7(2)=-N1*T1(2)/M7
199 S7(1)=S7(1)+T7(1)
200 S7(2)=S7(2)+T7(2)
```

BSSLY

```
201 N1=N1-2*M7
202 M7=M7+1
203 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
204 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
205 T7(1)=+N1*T1(1)/M7
206 T7(2)=+N1*T1(2)/M7
207 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 209
208 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 212
209 S7(3)=S7(3)+T7(1)
210 S7(4)=S7(4)+T7(2)
211 IF (M7<35) THEN 193
212 Q7(1)=Z0(1)-.5*PI*(N7+.5)
213 Q7(2)=Z0(2)
214 T1(1)=+COS(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
215 T1(2)=-SIN(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
216 T7(1)=S7(1)*T1(1)-S7(2)*T1(2)
217 T7(2)=S7(1)*T1(2)+S7(2)*T1(1)
218 T7(3)=S7(3)*T1(1)-S7(4)*T1(2)
219 T7(4)=S7(3)*T1(2)+S7(4)*T1(1)
220 T1(1)=+SIN(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
221 T1(2)=+COS(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
222 T7(1)=T7(1)-S7(3)*T1(1)+S7(4)*T1(2)
223 T7(2)=T7(2)-S7(3)*T1(2)-S7(4)*T1(1)
224 T7(3)=T7(3)+S7(1)*T1(1)-S7(2)*T1(2)
225 T7(4)=T7(4)+S7(1)*T1(2)+S7(2)*T1(1)
226 IF (Z(1)>=0) THEN 237
227 IF (Z(2)>=0) THEN 231
228 S7(1)=-2*T7(1)+T7(4)
229 S7(2)=-2*T7(2)-T7(3)
230 GOTO 233
231 S7(1)=-2*T7(1)-T7(4)
232 S7(2)=-2*T7(2)+T7(3)
233 IF (N7=0) THEN 239
234 S7(1)=-S7(1)
235 S7(2)=-S7(2)
236 GOTO 239
237 S7(1)=T7(3)
238 S7(2)=T7(4)
239 RETURN
240 S7(1)=1
241 S7(2)=0
242 S7(3)=1
243 S7(4)=0
244 J=15*N7+2
245 L=15*N7+15
246 FOR I=J TO L
247 T1(1)=+Z0(2)-D(I)
248 T1(2)=-Z0(1)
249 S2=T1(1)*T1(1)+T1(2)*T1(2)
250 T7(1)=+E(I)*T1(1)/S2
```

```
251 T7(2)=-E(I)*T1(2)/S2
252 S7(1)=S7(1)+T7(1)
253 S7(2)=S7(2)+T7(2)
254 T1(1)=-Z0(2)-D(I)
255 T1(2)=+Z0(1)
256 S2=T1(1)*T1(1)+T1(2)*T1(2)
257 T7(1)=+E(I)*T1(1)/S2
258 T7(2)=-E(I)*T1(2)/S2
259 S7(3)=S7(3)+T7(1)
260 S7(4)=S7(4)+T7(2)
261 NEXT I
262 Q7(1)=Z0(1)-.5*PI*(N7+.5)
263 Q7(2)=Z0(2)
264 T1(1)=+COS(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
265 T1(2)=-SIN(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
266 T7(1)=+T1(1)*S7(1)-T1(2)*S7(2)+T1(1)*S7(3)-T1(2)*S7(4)
267 T7(2)=+T1(1)*S7(2)+T1(2)*S7(1)+T1(1)*S7(4)+T1(2)*S7(3)
268 T7(3)=+T1(1)*S7(2)+T1(2)*S7(1)-T1(1)*S7(4)-T1(2)*S7(3)
269 T7(4)=-T1(1)*S7(1)+T1(2)*S7(2)+T1(1)*S7(3)-T1(2)*S7(4)
270 T1(1)=+SIN(Q7(1))*.5*(EXP(+Q7(2))+EXP(-Q7(2)))
271 T1(2)=+COS(Q7(1))*.5*(EXP(+Q7(2))-EXP(-Q7(2)))
272 T7(1)=T7(1)-T1(1)*S7(2)-T1(2)*S7(1)+T1(1)*S7(4)+T1(2)*S7(3)
273 T7(2)=T7(2)+T1(1)*S7(1)-T1(2)*S7(2)-T1(1)*S7(3)+T1(2)*S7(4)
274 T7(3)=T7(3)+T1(1)*S7(1)-T1(2)*S7(2)+T1(1)*S7(3)-T1(2)*S7(4)
275 T7(4)=T7(4)+T1(1)*S7(2)+T1(2)*S7(1)+T1(1)*S7(4)+T1(2)*S7(3)
276 IF (Z(1)>=0) THEN 287
277 IF (Z(2)>=0) THEN 281
278 S7(1)=-2*T7(1)+T7(4)
279 S7(2)=-2*T7(2)-T7(3)
280 GOTO 283
281 S7(1)=-2*T7(1)-T7(4)
282 S7(2)=-2*T7(2)+T7(3)
283 IF (N7=0) THEN 289
284 S7(1)=-S7(1)
285 S7(2)=-S7(2)
286 GOTO 289
287 S7(1)=T7(3)
288 S7(2)=T7(4)
289 RETURN
290 Q7(1)=1
291 Q7(2)=0
292 N2=0
293 M7=0
294 GOTO 301
295 M7=M7+1
296 N2=N2+.5/M7
297 T1(1)=.5*(Z(1)*Q7(1)-Z(2)*Q7(2))
298 T1(2)=.5*(Z(1)*Q7(2)+Z(2)*Q7(1))
299 Q7(1)=T1(1)/M7
300 Q7(2)=T1(2)/M7
```

```
301 IF (M7<N7) THEN 295
302 S1(1)=.25*(Z(1)-Z(2))*(Z(1)+Z(2))
303 S1(2)=.5*Z(1)*Z(2)
304 R2(1)=0
305 R2(2)=0
306 S2=Q7(1)*Q7(1)+Q7(2)*Q7(2)
307 T7(1)=+Q7(1)/S2
308 T7(2)=-Q7(2)/S2
309 M7=0
310 GOTO 320
311 T7(1)=T7(1)/(N7-M7)
312 T7(2)=T7(2)/(N7-M7)
313 R2(1)=R2(1)-.5*T7(1)
314 R2(2)=R2(2)-.5*T7(2)
315 M7=M7+1
316 T1(1)=S1(1)*T7(1)-S1(2)*T7(2)
317 T1(2)=S1(1)*T7(2)+S1(2)*T7(1)
318 T7(1)=+T1(1)/M7
319 T7(2)=+T1(2)/M7
320 IF (M7<N7) THEN 311
321 S7(1)=0
322 S7(2)=0
323 M2=1
324 M1=0
325 L7(1)=G-LOG(2)+L1(1)-N2
326 L7(2)=+L1(2)
327 M7=0
328 GOTO 337
329 M1=M1+M2
330 M7=M7+1
331 M2=.25*Z2*M2/(M7*(N7+M7))
332 T1(1)=S1(1)*Q7(1)-S1(2)*Q7(2)
333 T1(2)=S1(1)*Q7(2)+S1(2)*Q7(1)
334 Q7(1)=-T1(1)/(M7*(N7+M7))
335 Q7(2)=-T1(2)/(M7*(N7+M7))
336 L7(1)=L7(1)-.5/M7-.5/(N7+M7)
337 T7(1)=Q7(1)*L7(1)-Q7(2)*L7(2)
338 T7(2)=Q7(1)*L7(2)+Q7(2)*L7(1)
339 S7(1)=S7(1)+T7(1)
340 S7(2)=S7(2)+T7(2)
341 IF (M1+M2>M1) THEN 329
342 S7(1)=S7(1)+R2(1)
343 S7(2)=S7(2)+R2(2)
344 RETURN
345 Q1=ANGLE(X1,Y1)
346 RETURN
347 END SUB
348 OPTION NOLET
349 DIM Z(2),Y(2)
350 Z(1)=7
```

BSSLY

```
351 Z(2)=0
352 N=0
353 CALL BSSLY (Z(),N,Y())
354 PRINT USING "+#.##########^^^^": Y(1)
355 PRINT USING "+#.##########^^^^": Y(2)
356 N=1
357 CALL BSSLY (Z(),N,Y())
358 PRINT USING "+#.##########^^^^": Y(1)
359 PRINT USING "+#.##########^^^^": Y(2)
360 END
```

```
1 SUB BSSLI (M,Z(),N,I())
2 !********************************
3 !BASIC SUBROUTINE FOR BESSEL I
4 !********************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (INTEGER)
7 !I = FUNCTION (COMPLEX ARRAY)
8 !
9 OPTION NOLET
10 IF (H=1) THEN 88
11 H=1
12 DIM D0(15),D1(15),E0(15),E1(15)
13 DIM R0(2),Z0(2),R1(2),S1(2),T1(2)
14 DIM R7(4),S7(4),T7(2),Q7(2),F1(2)
15 DATA  0.000000000000000E00
16 DATA -1.648995051422117E-2
17 DATA -7.186218800685365E-2
18 DATA -1.670868781248656E-1
19 DATA -3.025822502194688E-1
20 DATA -4.806139452459267E-1
21 DATA -7.070752393578979E-1
22 DATA -9.929957905395160E-1
23 DATA -1.355839256125922E00
24 DATA -1.821059078991320E00
25 DATA -2.424821753108787E00
26 DATA -3.219566557087496E00
27 DATA -4.286580772483836E00
28 DATA -5.770228167981279E00
29 DATA -8.013712609525260E00
30 FOR K=1 TO 15
31 READ D0(K)
32 NEXT K
33 DATA  0.000000000000000E00
34 DATA -5.577424298795054E-3
35 DATA -4.991129441724757E-2
36 DATA -1.374409116523968E-1
37 DATA -2.672337847105657E-1
38 DATA -4.403801668086817E-1
39 DATA -6.618136148725406E-1
40 DATA -9.418610776650166E-1
41 DATA -1.297541304683261E00
42 DATA -1.754076967198164E00
43 DATA -2.347552998822763E00
44 DATA -3.130413326891964E00
45 DATA -4.183971205637291E00
46 DATA -5.652517992149936E00
47 DATA -7.878639598106769E00
48 FOR K=1 TO 15
49 READ D1(K)
50 NEXT K
```

```
51 DATA  0.000000000000000E00
52 DATA -4.809423363874473E-3
53 DATA -1.313662003477595E-2
54 DATA -1.948438340084579E-2
55 DATA -2.199489000320033E-2
56 DATA -2.093966256765194E-2
57 DATA -1.746002684586503E-2
58 DATA -1.279378133620848E-2
59 DATA -8.052344217965918E-3
60 DATA -4.158173750027601E-3
61 DATA -1.643177387479224E-3
62 DATA -4.491755853147087E-4
63 DATA -7.285947655740069E-5
64 DATA -5.382652306582855E-6
65 DATA -9.937790480362892E-8
66 FOR K=1 TO 15
67 READ E0(K)
68 NEXT K
69 DATA  0.000000000000000E00
70 DATA  7.538057792005914E-2
71 DATA  7.122935374034643E-2
72 DATA  6.331162242281997E-2
73 DATA  5.282402645233011E-2
74 DATA  4.133053594414916E-2
75 DATA  3.013505739475096E-2
76 DATA  2.010434395927201E-2
77 DATA  1.185522230680744E-2
78 DATA  5.860555109560099E-3
79 DATA  2.254651482673253E-3
80 DATA  6.081730415363355E-4
81 DATA  9.842155506257467E-5
82 DATA  7.321390930380890E-6
83 DATA  1.372796673846658E-7
84 FOR K=1 TO 15
85 READ E1(K)
86 NEXT K
87 RESTORE
88 Z2=Z(1)*Z(1)+Z(2)*Z(2)
89 Z1=SQR(Z2)
90 N0=ABS(N)
91 S0=+1
92 IF (Z(1)>=0) THEN 98
93 Z0(1)=-Z(1)
94 Z0(2)=-Z(2)
95 IF (N=2*IP(N/2)) THEN 100
96 S0=-1
97 GOTO 100
98 Z0(1)=+Z(1)
99 Z0(2)=+Z(2)
100 IF (Z1<=17.5+.5*N0*N0) THEN 103
```

```
101 N1=N0
102 GOTO 119
103 N1=.5*Z1-.5*ABS(Z0(1))+.5*ABS(.5*Z1-ABS(Z0(1)))
104 IF (N0<=N1) THEN 109
105 N1=+IP(.25*Z2)
106 IF (N0<=N1) THEN 294
107 N1=N0
108 GOTO 294
109 IF (Z1<=17.5) THEN 112
110 N1=+IP(SQR(2*(Z1-17.5)))
111 GOTO 119
112 IF (Z2<1) THEN 114
113 IF (-ABS(Z(1))+.096*Z(2)*Z(2)>=0) THEN 118
114 N1=+IP(.25*Z2)
115 IF (N0<=N1) THEN 294
116 N1=N0
117 GOTO 294
118 N1=0
119 S1(1)=Z0(1)
120 S1(2)=Z0(2)
121 M1=S0*SQR(1/(2*PI))
122 R0(1)=SQR(S1(1)+Z1)
123 R0(2)=S1(2)/R0(1)
124 R0(1)=SQR(.5)*R0(1)
125 R0(2)=SQR(.5)*R0(2)
126 F1(1)=+M1*R0(1)/Z1
127 F1(2)=-M1*R0(2)/Z1
128 IF (Z1<=17.5) THEN 166
129 R1(1)=+.5*Z0(1)/Z2
130 R1(2)=-.5*Z0(2)/Z2
131 N2=N1*N1-.25
132 S7(1)=0
133 S7(2)=0
134 S7(3)=0
135 S7(4)=0
136 T7(1)=1
137 T7(2)=0
138 M0=0
139 GOTO 146
140 N2=N2-2*M0
141 M0=M0+1
142 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
143 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
144 T7(1)=+N2*T1(1)/M0
145 T7(2)=+N2*T1(2)/M0
146 S7(1)=S7(1)+T7(1)
147 S7(2)=S7(2)+T7(2)
148 N2=N2-2*M0
149 M0=M0+1
150 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
```

```
151 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
152 T7(1)=+N2*T1(1)/M0
153 T7(2)=+N2*T1(2)/M0
154 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 156
155 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 159
156 S7(3)=S7(3)+T7(1)
157 S7(4)=S7(4)+T7(2)
158 IF (M0<35) THEN 140
159 T1(1)=S7(1)+S7(3)
160 T1(2)=S7(2)+S7(4)
161 S7(1)=S7(1)-S7(3)
162 S7(2)=S7(2)-S7(4)
163 S7(3)=T1(1)
164 S7(4)=T1(2)
165 GOTO 186
166 S7(1)=1
167 S7(2)=0
168 S7(3)=1
169 S7(4)=0
170 FOR K=2 TO 15
171 T1(1)=-Z0(1)-D0(K)
172 T1(2)=-Z0(2)
173 S2=T1(1)*T1(1)+T1(2)*T1(2)
174 T7(1)=+E0(K)*T1(1)/S2
175 T7(2)=-E0(K)*T1(2)/S2
176 S7(1)=S7(1)+T7(1)
177 S7(2)=S7(2)+T7(2)
178 T1(1)=+Z0(1)-D0(K)
179 T1(2)=+Z0(2)
180 S2=T1(1)*T1(1)+T1(2)*T1(2)
181 T7(1)=+E0(K)*T1(1)/S2
182 T7(2)=-E0(K)*T1(2)/S2
183 S7(3)=S7(3)+T7(1)
184 S7(4)=S7(4)+T7(2)
185 NEXT K
186 R7(1)=S7(1)
187 R7(2)=S7(2)
188 IF (Z0(1)>=17.5) THEN 199
189 Q7(1)=-2*Z0(1)
190 IF (Z0(2)>=0) THEN 193
191 Q7(2)=-2*Z0(2)-PI*(N1+.5)
192 GOTO 194
193 Q7(2)=-2*Z0(2)+PI*(N1+.5)
194 M1=EXP(Q7(1))
195 T1(1)=M1*COS(Q7(2))
196 T1(2)=M1*SIN(Q7(2))
197 R7(1)=R7(1)+T1(1)*S7(3)-T1(2)*S7(4)
198 R7(2)=R7(2)+T1(1)*S7(4)+T1(2)*S7(3)
199 IF (N1=N0) THEN 283
200 R7(3)=R7(1)
```

```
201 R7(4)=R7(2)
202 N1=N1+1
203 IF (Z1<=17.5) THEN 239
204 N2=N1*N1-.25
205 S7(1)=0
206 S7(2)=0
207 S7(3)=0
208 S7(4)=0
209 T7(1)=1
210 T7(2)=0
211 M0=0
212 GOTO 219
213 N2=N2-2*M0
214 M0=M0+1
215 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
216 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
217 T7(1)=+N2*T1(1)/M0
218 T7(2)=+N2*T1(2)/M0
219 S7(1)=S7(1)+T7(1)
220 S7(2)=S7(2)+T7(2)
221 N2=N2-2*M0
222 M0=M0+1
223 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
224 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
225 T7(1)=+N2*T1(1)/M0
226 T7(2)=+N2*T1(2)/M0
227 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 229
228 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 232
229 S7(3)=S7(3)+T7(1)
230 S7(4)=S7(4)+T7(2)
231 IF (M0<35) THEN 213
232 T1(1)=S7(1)+S7(3)
233 T1(2)=S7(2)+S7(4)
234 S7(1)=S7(1)-S7(3)
235 S7(2)=S7(2)-S7(4)
236 S7(3)=T1(1)
237 S7(4)=T1(2)
238 GOTO 259
239 S7(1)=1
240 S7(2)=0
241 S7(3)=1
242 S7(4)=0
243 FOR K=2 TO 15
244 T1(1)=-Z0(1)-D1(K)
245 T1(2)=-Z0(2)
246 S2=T1(1)*T1(1)+T1(2)*T1(2)
247 T7(1)=+E1(K)*T1(1)/S2
248 T7(2)=-E1(K)*T1(2)/S2
249 S7(1)=S7(1)+T7(1)
250 S7(2)=S7(2)+T7(2)
```

```
251 T1(1)=+Z0(1)-D1(K)
252 T1(2)=+Z0(2)
253 S2=T1(1)*T1(1)+T1(2)*T1(2)
254 T7(1)=+E1(K)*T1(1)/S2
255 T7(2)=-E1(K)*T1(2)/S2
256 S7(3)=S7(3)+T7(1)
257 S7(4)=S7(4)+T7(2)
258 NEXT K
259 R7(1)=S7(1)
260 R7(2)=S7(2)
261 IF (Z0(1)>=17.5) THEN 282
262 Q7(1)=-2*Z0(1)
263 IF (Z0(2)>=0) THEN 266
264 Q7(2)=-2*Z0(2)-PI*(N1+.5)
265 GOTO 267
266 Q7(2)=-2*Z0(2)+PI*(N1+.5)
267 M1=EXP(Q7(1))
268 T1(1)=M1*COS(Q7(2))
269 T1(2)=M1*SIN(Q7(2))
270 R7(1)=R7(1)+T1(1)*S7(3)-T1(2)*S7(4)
271 R7(2)=R7(2)+T1(1)*S7(4)+T1(2)*S7(3)
272 GOTO 282
273 T7(1)=-2*N1*Z0(1)/Z2
274 T7(2)=+2*N1*Z0(2)/Z2
275 T1(1)=T7(1)*R7(1)-T7(2)*R7(2)+R7(3)
276 T1(2)=T7(1)*R7(2)+T7(2)*R7(1)+R7(4)
277 R7(3)=R7(1)
278 R7(4)=R7(2)
279 R7(1)=T1(1)
280 R7(2)=T1(2)
281 N1=N1+1
282 IF (N1<N0) THEN 273
283 IF (M<>0) THEN 291
284 M1=EXP(Z0(1))
285 T7(1)=M1*COS(Z0(2))
286 T7(2)=M1*SIN(Z0(2))
287 T1(1)=T7(1)*R7(1)-T7(2)*R7(2)
288 T1(2)=T7(1)*R7(2)+T7(2)*R7(1)
289 R7(1)=T1(1)
290 R7(2)=T1(2)
291 I(1)=F1(1)*R7(1)-F1(2)*R7(2)
292 I(2)=F1(1)*R7(2)+F1(2)*R7(1)
293 EXIT SUB
294 S1(1)=+.25*(Z0(1)*Z0(1)-Z0(2)*Z0(2))
295 S1(2)=+.5*Z0(1)*Z0(2)
296 N2=N1
297 S7(1)=0
298 S7(2)=0
299 S7(3)=0
300 S7(4)=0
```

```
301 T7(1)=1
302 T7(2)=0
303 M0=0
304 N2=N2+1
305 T1(1)=+T7(1)/N2
306 T1(2)=+T7(2)/N2
307 S7(3)=S7(3)+T1(1)
308 S7(4)=S7(4)+T1(2)
309 T7(1)=+T1(1)*S1(1)-T1(2)*S1(2)
310 T7(2)=+T1(1)*S1(2)+T1(2)*S1(1)
311 M0=M0+1
312 T7(1)=T7(1)/M0
313 T7(2)=T7(2)/M0
314 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 316
315 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 319
316 S7(1)=S7(1)+T7(1)
317 S7(2)=S7(2)+T7(2)
318 GOTO 304
319 S7(1)=S7(1)+1
320 N2=N1+1
321 S7(3)=N2*S7(3)
322 S7(4)=N2*S7(4)
323 GOTO 334
324 N2=N1*(N1+1)
325 T7(1)=S1(1)/N2
326 T7(2)=S1(2)/N2
327 T1(1)=+T7(1)*S7(3)-T7(2)*S7(4)
328 T1(2)=+T7(1)*S7(4)+T7(2)*S7(3)
329 S7(3)=S7(1)
330 S7(4)=S7(2)
331 S7(1)=S7(1)+T1(1)
332 S7(2)=S7(2)+T1(2)
333 N1=N1-1
334 IF (N1>N0) THEN 324
335 F1(1)=S0
336 F1(2)=0
337 N1=0
338 GOTO 344
339 N1=N1+1
340 T7(1)=F1(1)*Z0(1)-F1(2)*Z0(2)
341 T7(2)=F1(1)*Z0(2)+F1(2)*Z0(1)
342 F1(1)=.5*T7(1)/N1
343 F1(2)=.5*T7(2)/N1
344 IF (N1<N0) THEN 339
345 IF (M=0) THEN 353
346 M1=EXP(-Z0(1))
347 T7(1)=M1*COS(-Z0(2))
348 T7(2)=M1*SIN(-Z0(2))
349 T1(1)=T7(1)*F1(1)-T7(2)*F1(2)
350 T1(2)=T7(1)*F1(2)+T7(2)*F1(1)
```

```
351 F1(1)=T1(1)
352 F1(2)=T1(2)
353 I(1)=F1(1)*S7(1)-F1(2)*S7(2)
354 I(2)=F1(1)*S7(2)+F1(2)*S7(1)
355 EXIT SUB
356 END SUB
357 OPTION NOLET
358 DIM Z(2),I(2),F(2)
359 N=0
360 Z(1)=7
361 Z(2)=0
362 M=1
363 CALL BSSLI (M,Z(),N,I())
364 PRINT USING "+#.##########^^^^": I(1)
365 PRINT USING "+#.##########^^^^": I(2)
366 M=0
367 CALL BSSLI (M,Z(),N,I())
368 F(1)=EXP(-Z(1))*COS(-Z(2))
369 F(2)=EXP(-Z(1))*SIN(-Z(2))
370 PRINT USING "+#.##########^^^^": F(1)*I(1)-F(2)*I(2)
371 PRINT USING "+#.##########^^^^": F(1)*I(2)+F(2)*I(1)
372 END
```

```
1 SUB BSSLK (M,Z(),N,K())
2 !*********************************
3 !BASIC SUBROUTINE FOR BESSEL K
4 !*********************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (INTEGER)
7 !K = FUNCTION (COMPLEX ARRAY)
8 !
9 OPTION NOLET
10 IF (H=1) THEN 89
11 H=1
12 DIM D(30),E(30),Z0(2),L1(2)
13 DIM R1(2),S1(2),T1(2),L7(2),S7(2)
14 DIM T7(2),Q2(2),R2(2),Q7(2),F1(2)
15 DATA 0.5772156649015329E00
16 READ G
17 DATA  0.000000000000000E00
18 DATA -1.648995051422117E-2
19 DATA -7.186218800685365E-2
20 DATA -1.670868781248656E-1
21 DATA -3.025822502194688E-1
22 DATA -4.806139452459267E-1
23 DATA -7.070752393578979E-1
24 DATA -9.929957905395160E-1
25 DATA -1.355839256125922E00
26 DATA -1.821059078991320E00
27 DATA -2.424821753108787E00
28 DATA -3.219566557087496E00
29 DATA -4.286580772483836E00
30 DATA -5.770228167981279E00
31 DATA -8.013712609525260E00
32 FOR I=1 TO 15
33 READ D(I)
34 NEXT I
35 DATA  0.000000000000000E00
36 DATA -5.577424298795054E-3
37 DATA -4.991129441724757E-2
38 DATA -1.374409116523968E-1
39 DATA -2.672337847105657E-1
40 DATA -4.403801668086817E-1
41 DATA -6.618136148725406E-1
42 DATA -9.418610776650166E-1
43 DATA -1.297541304683261E00
44 DATA -1.754076967198164E00
45 DATA -2.347552998822763E00
46 DATA -3.130413326891964E00
47 DATA -4.183971205637291E00
48 DATA -5.652517992149936E00
49 DATA -7.878639598106769E00
50 FOR I=16 TO 30
```

```
51 READ D(I)
52 NEXT I
53 DATA  0.000000000000000E00
54 DATA -4.809423363874473E-3
55 DATA -1.313662003477595E-2
56 DATA -1.948438340084579E-2
57 DATA -2.199489000320033E-2
58 DATA -2.093966256765194E-2
59 DATA -1.746002684586503E-2
60 DATA -1.279378133620848E-2
61 DATA -8.052344217965918E-3
62 DATA -4.158173750027601E-3
63 DATA -1.643177387479224E-3
64 DATA -4.491755853147087E-4
65 DATA -7.285947655740069E-5
66 DATA -5.382652306582855E-6
67 DATA -9.937790480362892E-8
68 FOR I=1 TO 15
69 READ E(I)
70 NEXT I
71 DATA  0.000000000000000E00
72 DATA  7.538057792005914E-2
73 DATA  7.122935374034643E-2
74 DATA  6.331162242281997E-2
75 DATA  5.282402645233011E-2
76 DATA  4.133053594414916E-2
77 DATA  3.013505739475096E-2
78 DATA  2.010434395927201E-2
79 DATA  1.185522230680744E-2
80 DATA  5.860555109560099E-3
81 DATA  2.254651482673253E-3
82 DATA  6.081730415363355E-4
83 DATA  9.842155506257467E-5
84 DATA  7.321390930380890E-6
85 DATA  1.372796673846658E-7
86 FOR I=16 TO 30
87 READ E(I)
88 NEXT I
89 RESTORE
90 Z2=Z(1)*Z(1)+Z(2)*Z(2)
91 L1(1)=.5*LOG(Z2)
92 X1=Z(1)
93 Y1=Z(2)
94 GOSUB 276
95 L1(2)=Q1
96 N0=ABS(N)
97 T7(1)=0
98 T7(2)=0
99 IF (M<>0) THEN 102
100 T7(1)=Z(1)
```

```
101 T7(2)=Z(2)
102 IF (Z2<=1) THEN 145
103 IF (Z2>=289) THEN 105
104 IF (Z(1)+.096*Z(2)*Z(2)<=0) THEN 145 ELSE 125
105 M1=SQR(PI/2)*EXP(-.5*L1(1)-T7(1))
106 F1(1)=M1*COS(-.5*L1(2)-T7(2))
107 F1(2)=M1*SIN(-.5*L1(2)-T7(2))
108 IF (NO>1) THEN 116
109 N7=NO
110 GOSUB 182
111 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
112 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
113 S7(1)=T1(1)
114 S7(2)=T1(2)
115 GOTO 179
116 N7=1
117 GOSUB 182
118 Q2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
119 Q2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
120 N7=0
121 GOSUB 182
122 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
123 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
124 GOTO 167
125 M1=SQR(PI/2)*EXP(-.5*L1(1)-T7(1))
126 F1(1)=M1*COS(-.5*L1(2)-T7(2))
127 F1(2)=M1*SIN(-.5*L1(2)-T7(2))
128 IF (NO>1) THEN 136
129 N7=NO
130 GOSUB 203
131 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
132 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
133 S7(1)=T1(1)
134 S7(2)=T1(2)
135 GOTO 179
136 N7=1
137 GOSUB 203
138 Q2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
139 Q2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
140 N7=0
141 GOSUB 203
142 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
143 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
144 GOTO 167
145 F1(1)=1
146 F1(2)=0
147 IF (M=0) THEN 151
148 M1=EXP(Z(1))
149 F1(1)=M1*COS(Z(2))
150 F1(2)=M1*SIN(Z(2))
```

```
151 IF (N0>1) THEN 159
152 N7=N0
153 GOSUB 217
154 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
155 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
156 S7(1)=T1(1)
157 S7(2)=T1(2)
158 GOTO 179
159 N7=1
160 GOSUB 217
161 Q2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
162 Q2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
163 N7=0
164 GOSUB 217
165 R2(1)=+F1(1)*S7(1)-F1(2)*S7(2)
166 R2(2)=+F1(1)*S7(2)+F1(2)*S7(1)
167 R1(1)=+Z(1)/Z2
168 R1(2)=-Z(2)/Z2
169 N7=0
170 GOTO 175
171 Q2(1)=R2(1)
172 Q2(2)=R2(2)
173 R2(1)=S7(1)
174 R2(2)=S7(2)
175 S7(1)=2*N7*(R1(1)*R2(1)-R1(2)*R2(2))+Q2(1)
176 S7(2)=2*N7*(R1(1)*R2(2)+R1(2)*R2(1))+Q2(2)
177 N7=N7+1
178 IF (N7<N0) THEN 171
179 K(1)=S7(1)
180 K(2)=S7(2)
181 EXIT SUB
182 S7(1)=0
183 S7(2)=0
184 R1(1)=+.5*Z(1)/Z2
185 R1(2)=-.5*Z(2)/Z2
186 N1=(N7-.5)*(N7+.5)
187 T7(1)=1
188 T7(2)=0
189 M7=0
190 GOTO 199
191 N1=N1-2*M7
192 M7=M7+1
193 T1(1)=T7(1)*R1(1)-T7(2)*R1(2)
194 T1(2)=T7(1)*R1(2)+T7(2)*R1(1)
195 T7(1)=+N1*T1(1)/M7
196 T7(2)=+N1*T1(2)/M7
197 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 199
198 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 202
199 S7(1)=S7(1)+T7(1)
200 S7(2)=S7(2)+T7(2)
```

```
201 IF (M7<36) THEN 191
202 RETURN
203 S7(1)=1
204 S7(2)=0
205 J=15*N7+2
206 L=15*N7+15
207 FOR I=J TO L
208 T1(1)=+Z(1)-D(I)
209 T1(2)=+Z(2)
210 S2=T1(1)*T1(1)+T1(2)*T1(2)
211 T7(1)=+E(I)*T1(1)/S2
212 T7(2)=-E(I)*T1(2)/S2
213 S7(1)=S7(1)+T7(1)
214 S7(2)=S7(2)+T7(2)
215 NEXT I
216 RETURN
217 Q7(1)=1
218 Q7(2)=0
219 N2=0
220 S0=-1
221 M7=0
222 GOTO 230
223 M7=M7+1
224 N2=N2+.5/M7
225 S0=-S0
226 T1(1)=.5*(Z(1)*Q7(1)-Z(2)*Q7(2))
227 T1(2)=.5*(Z(1)*Q7(2)+Z(2)*Q7(1))
228 Q7(1)=T1(1)/M7
229 Q7(2)=T1(2)/M7
230 IF (M7<N7) THEN 223
231 S1(1)=.25*(Z(1)-Z(2))*(Z(1)+Z(2))
232 S1(2)=.5*Z(1)*Z(2)
233 R2(1)=0
234 R2(2)=0
235 S2=Q7(1)*Q7(1)+Q7(2)*Q7(2)
236 T7(1)=+Q7(1)/S2
237 T7(2)=-Q7(2)/S2
238 M7=0
239 GOTO 249
240 T7(1)=T7(1)/(N7-M7)
241 T7(2)=T7(2)/(N7-M7)
242 R2(1)=R2(1)+.5*T7(1)
243 R2(2)=R2(2)+.5*T7(2)
244 M7=M7+1
245 T1(1)=S1(1)*T7(1)-S1(2)*T7(2)
246 T1(2)=S1(1)*T7(2)+S1(2)*T7(1)
247 T7(1)=-T1(1)/M7
248 T7(2)=-T1(2)/M7
249 IF (M7<N7) THEN 240
250 S7(1)=0
```

```
251 S7(2)=0
252 M2=1
253 M1=0
254 Q7(1)=S0*Q7(1)
255 Q7(2)=S0*Q7(2)
256 L7(1)=G-LOG(2)+L1(1)-N2
257 L7(2)=+L1(2)
258 M7=0
259 GOTO 268
260 M1=M1+M2
261 M7=M7+1
262 M2=.25*Z2*M2/(M7*(N7+M7))
263 T1(1)=S1(1)*Q7(1)-S1(2)*Q7(2)
264 T1(2)=S1(1)*Q7(2)+S1(2)*Q7(1)
265 Q7(1)=+T1(1)/(M7*(N7+M7))
266 Q7(2)=+T1(2)/(M7*(N7+M7))
267 L7(1)=L7(1)-.5/M7-.5/(N7+M7)
268 T7(1)=Q7(1)*L7(1)-Q7(2)*L7(2)
269 T7(2)=Q7(1)*L7(2)+Q7(2)*L7(1)
270 S7(1)=S7(1)+T7(1)
271 S7(2)=S7(2)+T7(2)
272 IF (M1+M2>M1) THEN 260
273 S7(1)=S7(1)+R2(1)
274 S7(2)=S7(2)+R2(2)
275 RETURN
276 Q1=ANGLE(X1,Y1)
277 RETURN
278 END SUB
279 OPTION NOLET
280 DIM Z(2),K(2),F(2)
281 N=0
282 Z(1)=7
283 Z(2)=0
284 M=1
285 CALL BSSLK (M,Z(),N,K())
286 PRINT USING "+#.##########^^^^": K(1)
287 PRINT USING "+#.##########^^^^": K(2)
288 M=0
289 CALL BSSLK (M,Z(),N,K())
290 F(1)=EXP(+Z(1))*COS(+Z(2))
291 F(2)=EXP(+Z(1))*SIN(+Z(2))
292 PRINT USING "+#.##########^^^^": F(1)*K(1)-F(2)*K(2)
293 PRINT USING "+#.##########^^^^": F(1)*K(2)+F(2)*K(1)
294 END
```

```
1 SUB CBSSLJ (Z(),N(),J())
2 !******************************
3 !COMPLEX BESSEL FUNCTION J
4 !******************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (COMPLEX ARRAY)
7 !J = FUNCTION (COMPLEX ARRAY)
8 OPTION NOLET
9 IF (H=1) THEN 14
10 H=1
11 DIM A0(2),G0(2),L0(2),Q0(2),R0(2),Z0(2),S1(2),Z1(2)
12 DIM T1(2),T7(2),R7(4),S7(4),R3(4),Q1(2),R1(2),F1(2)
13 DIM K0(2),U1(2),C0(16)
14 Z2=Z(1)*Z(1)+Z(2)*Z(2)
15 Z7=SQR(Z2)
16 N2=N(1)*N(1)+N(2)*N(2)
17 N0=IP(N(1))
18 F0=N(1)-N0
19 S0=+1
20 IF (F0<>0 OR N(2)<>0) THEN 25
21 N7=N0
22 N0=ABS(N0)
23 IF (N7>=0 OR N7=2*IP(N7/2)) THEN 25
24 S0=-1
25 IF (Z7>17.5+.5*N2) THEN 46
26 IF (Z7<=17.5) THEN 29
27 N1=N0
28 GOTO 115
29 IF (N0+F0>=0) THEN 41
30 M0=N(2)
31 M1=Z(2)-.5*N(2)
32 N1=-1.25*(Z7+.5*ABS(M0)-ABS(M1))
33 IF(N0+F0>=N1) THEN 41
34 M1=Z(2)-N(2)
35 N1=+1.25*Z7-.625*ABS(1.2*Z7-M1)-.625*ABS(1.2*Z7+M1)
36 IF (N0+F0>=N1) THEN 41
37 N1=-IP(1.25*(Z7-ABS(M0)))
38 IF (N0>=N1) THEN 187
39 N1=N0
40 GOTO 187
41 M1=.0625*Z2*Z2-N(2)*N(2)
42 N1=+IP(SQR(.5*(M1+ABS(M1))))
43 IF (N0<N1) THEN 187
44 N1=N0
45 GOTO 187
46 IF (Z(1)>=0) THEN 59
47 Q0(1)=-Z(1)
48 Q0(2)=-Z(2)
49 IF (Z(2)>=0) THEN 53
50 A0(1)=+PI*N(2)
```

```
51 A0(2)=-PI*(N0+F0)
52 GOTO 55
53 A0(1)=-PI*N(2)
54 A0(2)=+PI*(N0+F0)
55 M1=S0*SQR(2/PI)*EXP(A0(1))
56 T7(1)=M1*COS(A0(2))
57 T7(2)=M1*SIN(A0(2))
58 GOTO 63
59 Q0(1)=+Z(1)
60 Q0(2)=+Z(2)
61 T7(1)=S0*SQR(2/PI)
62 T7(2)=0
63 Z0(1)=SQR(Q0(1)+Z7)
64 Z0(2)=Q0(2)/Z0(1)
65 Z0(1)=SQR(.5)*Z0(1)
66 Z0(2)=SQR(.5)*Z0(2)
67 F1(1)=+(T7(1)*Z0(1)+T7(2)*Z0(2))/Z7
68 F1(2)=-(T7(1)*Z0(2)-T7(2)*Z0(1))/Z7
69 R0(1)=+.5*Q0(1)/Z2
70 R0(2)=-.5*Q0(2)/Z2
71 T1(1)=N0+F0
72 T1(2)=N(2)
73 A0(1)=T1(1)*T1(1)-T1(2)*T1(2)-.25
74 A0(2)=2*T1(1)*T1(2)
75 S7(1)=0
76 S7(2)=0
77 S7(3)=0
78 S7(4)=0
79 T7(1)=1
80 T7(2)=0
81 M0=0
82 GOTO 89
83 A0(1)=A0(1)-2*M0
84 M0=M0+1
85 T1(1)=T7(1)*R0(1)-T7(2)*R0(2)
86 T1(2)=T7(1)*R0(2)+T7(2)*R0(1)
87 T7(1)=-(T1(1)*A0(1)-T1(2)*A0(2))/M0
88 T7(2)=-(T1(1)*A0(2)+T1(2)*A0(1))/M0
89 S7(1)=S7(1)+T7(1)
90 S7(2)=S7(2)+T7(2)
91 A0(1)=A0(1)-2*M0
92 M0=M0+1
93 T1(1)=T7(1)*R0(1)-T7(2)*R0(2)
94 T1(2)=T7(1)*R0(2)+T7(2)*R0(1)
95 T7(1)=+(T1(1)*A0(1)-T1(2)*A0(2))/M0
96 T7(2)=+(T1(1)*A0(2)+T1(2)*A0(1))/M0
97 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 99
98 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 102
99  S7(3)=S7(3)+T7(1)
100 S7(4)=S7(4)+T7(2)
```

```
101 IF (M0<35) THEN 83
102 A0(1)=Q0(1)-(PI/2)*(N0+F0+.5)
103 A0(2)=Q0(2)-(PI/2)*N(2)
104 T1(1)=+COS(A0(1))*.5*(EXP(+A0(2))+EXP(-A0(2)))
105 T1(2)=-SIN(A0(1))*.5*(EXP(+A0(2))-EXP(-A0(2)))
106 T7(1)=S7(1)*T1(1)-S7(2)*T1(2)
107 T7(2)=S7(1)*T1(2)+S7(2)*T1(1)
108 T1(1)=+SIN(A0(1))*.5*(EXP(+A0(2))+EXP(-A0(2)))
109 T1(2)=+COS(A0(1))*.5*(EXP(+A0(2))-EXP(-A0(2)))
110 R7(1)=T7(1)-S7(3)*T1(1)+S7(4)*T1(2)
111 R7(2)=T7(2)-S7(3)*T1(2)-S7(4)*T1(1)
112 J(1)=F1(1)*R7(1)-F1(2)*R7(2)
113 J(2)=F1(1)*R7(2)+F1(2)*R7(1)
114 EXIT SUB
115 N7=1
116 IF (ABS(N(2))>=.8*ABS(Z(2))) THEN N7=0
117 M0=N(2)
118 M1=Z(2)-.5*N(2)
119 M1=-1.25*(Z7+.5*ABS(M0)-ABS(M1))
120 IF (N0+F0>=M1) THEN N7=0
121 M1=Z(2)-N(2)
122 M1=+1.25*Z7-.625*ABS(1.2*Z7-M1)-.625*ABS(1.2*Z7+M1)
123 IF (N0+F0>=M1) THEN N7=0
124 IF (Z(1)>=0) THEN 137
125 Q0(1)=-Z(1)
126 Q0(2)=-Z(2)
127 IF (Z(2)>=0) THEN 131
128 A0(1)=+PI*N(2)
129 A0(2)=-PI*(N0+F0)
130 GOTO 133
131 A0(1)=-PI*N(2)
132 A0(2)=+PI*(N0+F0)
133 M1=S0/SQR(2*PI)*EXP(A0(1))
134 F1(1)=M1*COS(A0(2))
135 F1(2)=M1*SIN(A0(2))
136 GOTO 141
137 Q0(1)=+Z(1)
138 Q0(2)=+Z(2)
139 F1(1)=S0/SQR(2*PI)
140 F1(2)=0
141 A0(1)=N1+F0
142 A0(2)=N(2)
143 Z1(1)=+(A0(1)*Q0(1)+A0(2)*Q0(2))/Z2
144 Z1(2)=-(A0(1)*Q0(2)-A0(2)*Q0(1))/Z2
145 R7(1)=1-Z1(1)
146 R7(2)=-Z1(2)
147 R7(3)=1+Z1(1)
148 R7(4)=+Z1(2)
149 T7(1)=R7(1)*R7(1)+R7(2)*R7(2)
150 T7(2)=R7(3)*R7(3)+R7(4)*R7(4)
```

```
151 T1(1)=T7(1)*T7(1)
152 T1(2)=T7(2)*T7(2)
153 Q3=T7(1)*T1(1)*T7(2)*T1(2)
154 Q2=(T1(1)+T1(2))*(T1(1)+T1(2))
155 S2=16-6*Z2*Z2*Q3/Q2
156 IF (S2>1) THEN 163
157 N1=N1+1
158 IF (N7=0) THEN 141
159 N1=-IP(1.25*(Z7-ABS(M0)))
160 IF (N0>=N1) THEN 187
161 N1=N0
162 GOTO 187
163 A0(1)=N1+F0
164 A0(2)=N(2)
165 GOSUB 292
166 S7(1)=S7(3)
167 S7(2)=S7(4)
168 IF (N1=N0) THEN 184
169 A0(1)=1+N1+F0
170 A0(2)=N(2)
171 GOSUB 292
172 A0(1)=N1+F0
173 A0(2)=N(2)
174 T7(1)=+2*(A0(1)*Q0(1)+A0(2)*Q0(2))/Z2
175 T7(2)=-2*(A0(1)*Q0(2)-A0(2)*Q0(1))/Z2
176 T1(1)=T7(1)*S7(1)-T7(2)*S7(2)-S7(3)
177 T1(2)=T7(1)*S7(2)+T7(2)*S7(1)-S7(4)
178 S7(3)=S7(1)
179 S7(4)=S7(2)
180 S7(1)=T1(1)
181 S7(2)=T1(2)
182 N1=N1-1
183 IF (N1<>N0) THEN 172
184 J(1)=F1(1)*S7(1)-F1(2)*S7(2)
185 J(2)=F1(1)*S7(2)+F1(2)*S7(1)
186 EXIT SUB
187 M0=1
188 A0(1)=1+N0+F0
189 A0(2)=N(2)
190 CALL CGAMMA (M0,A0(),G0())
191 S1(1)=+.25*(Z(1)*Z(1)-Z(2)*Z(2))
192 S1(2)=+.5*Z(1)*Z(2)
193 M1=S1(1)*S1(1)+S1(2)*S1(2)
194 A0(1)=N1+F0
195 S7(1)=0
196 S7(2)=0
197 S7(3)=0
198 S7(4)=0
199 T7(1)=1
200 T7(2)=0
```

```
201 M0=0
202 IF (N1>=0) THEN 242
203 S2=A0(1)*A0(1)+A0(2)*A0(2)
204 T1(1)=+(T7(1)*A0(1)+T7(2)*A0(2))/S2
205 T1(2)=-(T7(1)*A0(2)-T7(2)*A0(1))/S2
206 S7(1)=S7(1)+T1(1)
207 S7(2)=S7(2)+T1(2)
208 T7(1)=-T1(1)*S1(1)+T1(2)*S1(2)
209 T7(2)=-T1(1)*S1(2)-T1(2)*S1(1)
210 M0=M0+1
211 T7(1)=T7(1)/M0
212 T7(2)=T7(2)/M0
213 IF (A0(1)<0) OR (M1>M0*M0*S2) THEN 216
214 IF (ABS(S7(3))+ABS(T7(1))>ABS(S7(3))) THEN 216
215 IF (ABS(S7(4))+ABS(T7(2))=ABS(S7(4))) THEN 220
216 S7(3)=S7(3)+T7(1)
217 S7(4)=S7(4)+T7(2)
218 A0(1)=A0(1)+1
219 GOTO 203
220 S7(3)=S7(3)+1
221 A0(1)=N1+F0
222 T1(1)=A0(1)*S7(1)-A0(2)*S7(2)
223 T1(2)=A0(1)*S7(2)+A0(2)*S7(1)
224 S7(1)=T1(1)
225 S7(2)=T1(2)
226 GOTO 240
227 A0(1)=N1+F0
228 T1(1)=A0(1)*A0(1)-A0(2)*A0(2)+A0(1)
229 T1(2)=2*A0(1)*A0(2)+A0(2)
230 S2=S1(1)*S1(1)+S1(2)*S1(2)
231 T7(1)=+(S1(1)*T1(1)+S1(2)*T1(2))/S2
232 T7(2)=+(S1(1)*T1(2)-S1(2)*T1(1))/S2
233 T1(1)=T7(1)*(S7(1)-S7(3))-T7(2)*(S7(2)-S7(4))
234 T1(2)=T7(1)*(S7(2)-S7(4))+T7(2)*(S7(1)-S7(3))
235 S7(3)=S7(1)
236 S7(4)=S7(2)
237 S7(1)=T1(1)
238 S7(2)=T1(2)
239 N1=N1+1
240 IF (N1<N0) THEN 227
241 GOTO 279
242 A0(1)=A0(1)+1
243 S2=A0(1)*A0(1)+A0(2)*A0(2)
244 T1(1)=+(T7(1)*A0(1)+T7(2)*A0(2))/S2
245 T1(2)=-(T7(1)*A0(2)-T7(2)*A0(1))/S2
246 S7(3)=S7(3)+T1(1)
247 S7(4)=S7(4)+T1(2)
248 T7(1)=-T1(1)*S1(1)+T1(2)*S1(2)
249 T7(2)=-T1(1)*S1(2)-T1(2)*S1(1)
250 M0=M0+1
```

```
251 T7(1)=T7(1)/M0
252 T7(2)=T7(2)/M0
253 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 255
254 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 258
255 S7(1)=S7(1)+T7(1)
256 S7(2)=S7(2)+T7(2)
257 GOTO 242
258 S7(1)=S7(1)+1
259 A0(1)=1+N1+F0
260 T1(1)=A0(1)*S7(3)-A0(2)*S7(4)
261 T1(2)=A0(1)*S7(4)+A0(2)*S7(3)
262 S7(3)=T1(1)
263 S7(4)=T1(2)
264 GOTO 278
265 A0(1)=N1+F0
266 T1(1)=A0(1)*A0(1)-A0(2)*A0(2)+A0(1)
267 T1(2)=2*A0(1)*A0(2)+A0(2)
268 S2=T1(1)*T1(1)+T1(2)*T1(2)
269 T7(1)=+(S1(1)*T1(1)+S1(2)*T1(2))/S2
270 T7(2)=-(S1(1)*T1(2)-S1(2)*T1(1))/S2
271 T1(1)=-T7(1)*S7(3)+T7(2)*S7(4)
272 T1(2)=-T7(1)*S7(4)-T7(2)*S7(3)
273 S7(3)=S7(1)
274 S7(4)=S7(2)
275 S7(1)=S7(1)+T1(1)
276 S7(2)=S7(2)+T1(2)
277 N1=N1-1
278 IF (N1>N0) THEN 265
279 A0(1)=N0+F0
280 L0(1)=.5*LOG(Z2)-LOG(2)
281 L0(2)=ANGLE(Z(1),Z(2))
282 T7(1)=A0(1)*L0(1)-A0(2)*L0(2)
283 T7(2)=A0(1)*L0(2)+A0(2)*L0(1)
284 T7(1)=T7(1)-G0(1)
285 T7(2)=T7(2)-G0(2)
286 M1=S0*EXP(T7(1))
287 F1(1)=M1*COS(T7(2))
288 F1(2)=M1*SIN(T7(2))
289 J(1)=F1(1)*S7(1)-F1(2)*S7(2)
290 J(2)=F1(1)*S7(2)+F1(2)*S7(1)
291 EXIT SUB
292 N2=A0(1)*A0(1)+A0(2)*A0(2)
293 Z1(1)=(A0(1)*Q0(1)+A0(2)*Q0(2))/N2
294 Z1(2)=(A0(1)*Q0(2)-A0(2)*Q0(1))/N2
295 S1(1)=(1-Z1(1))*(1+Z1(1))+Z1(2)*Z1(2)
296 S1(2)=-2*Z1(1)*Z1(2)
297 S2=S1(1)*S1(1)+S1(2)*S1(2)
298 R2=SQR(S2)
299 R0(1)=0
300 R0(2)=0
```

```
301 IF (S1(1)=0) THEN 308
302 IF (S1(1)>0) THEN 309
303 R0(2)=SQR(-S1(1)+R2)
304 R0(1)=S1(2)/R0(2)
305 IF (S1(2)>=0) THEN 311
306 M1=-1/SQR(2)
307 GOTO 312
308 IF (S1(2)=0) THEN 314
309 R0(1)=SQR(+S1(1)+R2)
310 R0(2)=S1(2)/R0(1)
311 M1=+1/SQR(2)
312 R0(1)=M1*R0(1)
313 R0(2)=M1*R0(2)
314 T1(1)=1+R0(1)
315 T1(2)=R0(2)
316 Q2=T1(1)*T1(1)+T1(2)*T1(2)
317 T7(1)=(T1(1)*Z1(1)+T1(2)*Z1(2))/Q2
318 T7(2)=(T1(1)*Z1(2)-T1(2)*Z1(1))/Q2
319 Q2=T7(1)*T7(1)+T7(2)*T7(2)
320 L0(1)=R0(1)+.5*LOG(Q2)
321 L0(2)=R0(2)+ANGLE(T7(1),T7(2))
322 T1(1)=A0(1)*L0(1)-A0(2)*L0(2)
323 T1(2)=A0(1)*L0(2)+A0(2)*L0(1)
324 Q1(1)=+S1(1)/S2
325 Q1(2)=-S1(2)/S2
326 A3=1/SQR(N2*R2)
327 A2=-1/R2
328 A1=1
329 T7(1)=A0(1)*R0(1)-A0(2)*R0(2)
330 T7(2)=A0(1)*R0(2)+A0(2)*R0(1)
331 Q2=T7(1)*T7(1)+T7(2)*T7(2)
332 S1(1)=+T7(1)/Q2
333 S1(2)=-T7(2)/Q2
334 S2=S1(1)*S1(1)+S1(2)*S1(2)
335 R2=SQR(S2)
336 R0(1)=0
337 R0(2)=0
338 IF (S1(1)=0) THEN 345
339 IF (S1(1)>0) THEN 346
340 R0(2)=SQR(-S1(1)+R2)
341 R0(1)=S1(2)/R0(2)
342 IF (S1(2)>=0) THEN 348
343 M1=-1/SQR(2)
344 GOTO 349
345 IF (S1(2)=0) THEN 351
346 R0(1)=SQR(+S1(1)+R2)
347 R0(2)=S1(2)/R0(1)
348 M1=+1/SQR(2)
349 R0(1)=M1*R0(1)
350 R0(2)=M1*R0(2)
```

```
351 R1(1)=S1(1)
352 R1(2)=S1(2)
353 M1=(1/3)*ANGLE(Q0(1),Q0(2))
354 T7(1)=COS(M1)
355 T7(2)=SIN(M1)
356 M1=.5*SQR(3)
357 R7(1)=+M1*T7(1)+.5*T7(2)
358 R7(2)=+M1*T7(2)-.5*T7(1)
359 R7(3)=+M1*T7(1)-.5*T7(2)
360 R7(4)=+M1*T7(2)+.5*T7(1)
361 K0(1)=1
362 K0(2)=0
363 IF ((A0(1)-Q0(1))*R7(1)+(A0(2)-Q0(2))*R7(2)<=0) THEN 365
364 K0(1)=0
365 IF ((A0(1)-Q0(1))*R7(3)+(A0(2)-Q0(2))*R7(4)<=0) THEN 367
366 K0(1)=0
367 IF ((A0(1)+Q0(1))*R7(1)+(A0(2)+Q0(2))*R7(2)>=0) THEN 369
368 K0(2)=1
369 IF ((A0(1)+Q0(1))*R7(3)+(A0(2)+Q0(2))*R7(4)>=0) THEN 371
370 K0(2)=1
371 Q2=A0(1)*T7(2)-A0(2)*T7(1)+Q0(1)*T7(2)-Q0(2)*T7(1)
372 IF (Q2<=0) THEN 374
373 K0(2)=-K0(2)
374 R7(1)=0
375 R7(2)=0
376 R7(3)=0
377 R7(4)=0
378 M1=EXP(T1(1))
379 T7(1)=M1*COS(T1(2))
380 T7(2)=M1*SIN(T1(2))
381 R3(1)=+T7(1)*R0(1)-T7(2)*R0(2)
382 R3(2)=+T7(1)*R0(2)+T7(2)*R0(1)
383 IF (K0(1)<>0) THEN 387
384 R3(3)=0
385 R3(4)=0
386 GOTO 397
387 Q2=T7(1)*T7(1)+T7(2)*T7(2)
388 T7(1)=+T7(1)/Q2
389 T7(2)=-T7(2)/Q2
390 R3(3)=+T7(1)*R0(2)+T7(2)*R0(1)
391 R3(4)=-T7(1)*R0(1)+T7(2)*R0(2)
392 IF (R1(2)<0) THEN 395
393 IF (R1(2)>0) THEN 397
394 IF (R1(1)<=0) THEN 397
395 R3(3)=-R3(3)
396 R3(4)=-R3(4)
397 FOR K7=1 TO 16
398 C0(K7)=0
399 NEXT K7
400 C0(1)=1
```

```
401 M7=1
402 GOTO 413
403 K2=0
404 M0=M7-1
405 FOR K7=1 TO M7
406 M2=K2
407 K2=C0(K7)
408 K1=(.125/M0+.5*(M0-1))*K2
409 M1=(.625/M0+.5*(M0-3))*M2
410 C0(K7)=K1-M1
411 M0=M0+2
412 NEXT K7
413 U1(1)=0
414 U1(2)=0
415 U0=0
416 L7=M7
417 FOR K7=1 TO M7
418 U0=C0(L7)+A2*U0
419 T1(1)=U1(1)*Q1(1)-U1(2)*Q1(2)
420 T1(2)=U1(1)*Q1(2)+U1(2)*Q1(1)
421 U1(1)=T1(1)+C0(L7)
422 U1(2)=T1(2)
423 L7=L7-1
424 NEXT K7
425 R7(1)=R7(1)+R3(1)*U1(1)-R3(2)*U1(2)
426 R7(2)=R7(2)+R3(1)*U1(2)+R3(2)*U1(1)
427 T1(1)=R3(1)*R1(1)-R3(2)*R1(2)
428 T1(2)=R3(1)*R1(2)+R3(2)*R1(1)
429 R3(1)=+T1(1)
430 R3(2)=+T1(2)
431 IF (K0(1)=0) THEN 438
432 R7(3)=R7(3)+R3(3)*U1(1)-R3(4)*U1(2)
433 R7(4)=R7(4)+R3(3)*U1(2)+R3(4)*U1(1)
434 T1(1)=R3(3)*R1(1)-R3(4)*R1(2)
435 T1(2)=R3(3)*R1(2)+R3(4)*R1(1)
436 R3(3)=-T1(1)
437 R3(4)=-T1(2)
438 U0=A1*U0
439 A1=A3*A1
440 IF (1+U0=1) THEN 443
441 M7=M7+1
442 IF (M7<=16) THEN 403
443 IF (K0(1)<>0) THEN 447
444 S7(3)=R7(1)
445 S7(4)=R7(2)
446 GOTO 484
447 IF (K0(2)<>0) THEN 451
448 S7(3)=R7(1)+R7(3)
449 S7(4)=R7(2)+R7(4)
450 GOTO 484
```

```
451 IF (Z1(2)>=0) THEN 455
452 T7(1)=-2*PI*A0(2)
453 T7(2)=+2*PI*F0
454 GOTO 457
455 T7(1)=+2*PI*A0(2)
456 T7(2)=-2*PI*F0
457 M1=EXP(T7(1))
458 T1(1)=M1*COS(T7(2))
459 T1(2)=M1*SIN(T7(2))
460 IF (Q0(2)<=0) AND (K0(2)<=0) THEN 469
461 IF (Q0(2)>=0) AND (K0(2)>=0) THEN 469
462 IF (Q0(2)<=0) AND  (Z1(2)<0) THEN 469
463 IF (Q0(2)>=0) AND (Z1(2)>=0) THEN 469
464 Q2=T1(1)*T1(1)+T1(2)*T1(2)
465 T7(1)=(T1(1)*R7(3)+T1(2)*R7(4))/Q2
466 T7(2)=(T1(1)*R7(4)-T1(2)*R7(3))/Q2
467 R7(3)=T7(1)
468 R7(4)=T7(2)
469 T7(1)=R7(1)-T1(1)*R7(1)+T1(2)*R7(2)
470 T7(2)=R7(2)-T1(1)*R7(2)-T1(2)*T7(1)
471 IF (Q0(1)<>0) THEN 475
472 IF (Q0(2)<0) AND  (A0(2)>0) THEN 482
473 IF (Q0(2)>0) AND (A0(2)<=0) THEN 482
474 GOTO 479
475 IF (Q0(2)<0)  AND  (Z1(2)<0) THEN 479
476 IF (Q0(2)>=0) AND (Z1(2)>=0) THEN 479
477 IF (Q0(2)<0)  AND (A0(2)>=0) AND (R1(2)<0)  THEN 482
478 IF (Q0(2)>=0) AND  (A0(2)<O) AND (R1(2)>=0) THEN 482
479 S7(3)=R7(3)+T7(1)
480 S7(4)=R7(4)+T7(2)
481 GOTO 484
482 S7(3)=R7(3)-T7(1)
483 S7(4)=R7(4)-T7(2)
484 RETURN
485 END SUB
486 SUB CGAMMA (M,Z(),G())
487 !******************************
488 !SUBROUTINE FOR GAMMA FUNCTION
489 !******************************
490 !M = MODE OF OPERATION
491 !Z = ARGUMENT (COMPLEX ARRAY)
492 !G = FUNCTION (COMPLEX ARRAY)
493 OPTION NOLET
494 IF (E=1) THEN 550
495 E=1
496 DIM C(26),B(18)
497 DIM F2(2)
498 DIM L1(2),L2(2),L3(2)
499 DATA +1.0000000000000000E0
500 DATA +0.5772156649015329E0
```

```
501 DATA -0.6558780715202538E0
502 DATA -0.0420026350340952E0
503 DATA +0.1665386113822915E0
504 DATA -0.0421977345555443E0
505 DATA -0.0096219715278770E0
506 DATA +0.0072189432466630E0
507 DATA -0.0011651675918591E0
508 DATA -0.0002152416741149E0
509 DATA +0.0001280502823882E0
510 DATA -0.0000201348547807E0
511 DATA -0.0000012504934821E0
512 DATA +0.0000011330272320E0
513 DATA -0.0000002056338417E0
514 DATA +0.0000000061160950E0
515 DATA +0.0000000050020075E0
516 DATA -0.0000000011812746E0
517 DATA +0.0000000001043427E0
518 DATA +0.0000000000077823E0
519 DATA -0.0000000000036968E0
520 DATA +0.0000000000005100E0
521 DATA -0.0000000000000207E0
522 DATA -0.0000000000000054E0
523 DATA +0.0000000000000014E0
524 DATA +0.0000000000000001E0
525 FOR I=1 TO 26
526 READ C(I)
527 NEXT I
528 DATA +8.333333333333333E-2
529 DATA -2.777777777777777E-3
530 DATA +7.936507936507937E-4
531 DATA -5.952380952380952E-4
532 DATA +8.417508417508418E-4
533 DATA -1.917526917526918E-3
534 DATA +6.410256410256410E-3
535 DATA -2.955065359477124E-2
536 DATA +1.796443723688306E-1
537 DATA -1.392432216905901E00
538 DATA +1.340286404416839E+1
539 DATA -1.568482846260020E+2
540 DATA +2.193103333333333E+3
541 DATA -3.610877125372499E+4
542 DATA +6.914722688513131E+5
543 DATA -1.523822153940742E+7
544 DATA +3.829007513914141E+8
545 DATA -1.088226603578439E10
546 FOR I=1 TO 18
547 READ B(I)
548 NEXT I
549 RESTORE
550 P1=IP(ABS(Z(1))+.5)
```

```
551 P1=ABS(P1)*SGN(Z(1))
552 IF (P1>0) THEN 586
553 X1=PI*(Z(1)-P1)
554 Y1=PI*Z(2)
555 S1(1)=SIN(X1)*.5*(EXP(+Y1)+EXP(-Y1))
556 S1(2)=COS(X1)*.5*(EXP(+Y1)-EXP(-Y1))
557 X1=1-Z(1)
558 Y1=-Z(2)
559 Q2=1-P1
560 GOSUB 593
561 IF (M<>0) THEN 573
562 N7=P1
563 Q2=2*IP(N7/2)
564 IF (P1=Q7) THEN 567
565 S1(1)=-S1(1)
566 S1(2)=-S1(2)
567 T1(1)=S1(1)*U-S1(2)*V
568 T1(2)=S1(1)*V+S1(2)*U
569 S2=T1(1)*T1(1)+T1(2)*T1(2)
570 G(1)=+PI*T1(1)/S2
571 G(2)=-PI*T1(2)/S2
572 EXIT SUB
573 S2=S1(1)*S1(1)+S1(2)*S1(2)
574 X=S1(1)
575 Y=S1(2)
576 GOSUB 680
577 L2(1)=.5*LOG(S2)
578 L2(2)=Q
579 IF (Z(2)>=0) THEN 582
580 L2(2)=L2(2)+PI*P1
581 GOTO 583
582 L2(2)=L2(2)-PI*P1
583 G(1)=LOG(PI)-L2(1)-U
584 G(2)=-L2(2)-V
585 EXIT SUB
586 X1=Z(1)
587 Y1=Z(2)
588 Q2=P1
589 GOSUB 593
590 G(1)=U
591 G(2)=V
592 EXIT SUB
593 L1(1)=0
594 L1(2)=0
595 S2=X1*X1+Y1*Y1
596 IF (S2>=32) THEN 611
597 F2(1)=1
598 F2(2)=0
599 X2=X1-Q7
600 Y2=Y1
```

```
601 T2=X2*X2+Y2*Y2
602 IF (T2<=1) THEN 649
603 X=X1
604 Y=Y1
605 GOSUB 680
606 L1(1)=L1(1)+.5*LOG(S2)
607 L1(2)=L1(2)+Q
608 X1=X1+1
609 S2=X1*X1+Y1*Y1
610 IF (S2<32) THEN 603
611 X=X1
612 Y=Y1
613 GOSUB 680
614 L3(1)=.5*LOG(S2)
615 L3(2)=Q
616 R1(1)=(X1-.5)*L3(1)-Y1*L3(2)-X1+.5*LOG(2*PI)
617 R1(2)=(X1-.5)*L3(2)+Y1*L3(1)-Y1
618 T7(1)=+X1/S2
619 T7(2)=-Y1/S2
620 F2(1)=T7(1)*T7(1)-T7(2)*T7(2)
621 F2(2)=2*T7(1)*T7(2)
622 S7(1)=0
623 S7(2)=0
624 FOR M7=1 TO 18
625 T1(1)=B(M7)*T7(1)
626 T1(2)=B(M7)*T7(2)
627 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 629
628 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 636
629 S7(1)=S7(1)+T1(1)
630 S7(2)=S7(2)+T1(2)
631 T1(1)=T7(1)*F2(1)-T7(2)*F2(2)
632 T1(2)=T7(1)*F2(2)+T7(2)*F2(1)
633 T7(1)=T1(1)
634 T7(2)=T1(2)
635 NEXT M7
636 U=R1(1)+S7(1)-L1(1)
637 V=R1(2)+S7(2)-L1(2)
638 IF (M<>0) THEN 644
639 Q2=EXP(U)
640 F2(1)=Q2*COS(V)
641 F2(2)=Q2*SIN(V)
642 U=F2(1)
643 V=F2(2)
644 RETURN
645 T1(1)=F2(1)*X1-F2(2)*Y1
646 T1(2)=F2(1)*Y1+F2(2)*X1
647 F2(1)=T1(1)
648 F2(2)=T1(2)
649 X1=X1-1
650 IF (X1>X2) THEN 645
```

```
651 T7(1)=1
652 T7(2)=0
653 S7(1)=0
654 S7(2)=0
655 FOR M7=1 TO 26
656 T1(1)=C(M7)*T7(1)
657 T1(2)=C(M7)*T7(2)
658 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 660
659 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 667
660 S7(1)=S7(1)+T1(1)
661 S7(2)=S7(2)+T1(2)
662 T1(1)=X1*T7(1)-Y1*T7(2)
663 T1(2)=X1*T7(2)+Y1*T7(1)
664 T7(1)=T1(1)
665 T7(2)=T1(2)
666 NEXT M7
667 S2=S7(1)*S7(1)+S7(2)*S7(2)
668 U=+(F2(1)*S7(1)+F2(2)*S7(2))/S2
669 V=-(F2(1)*S7(2)-F2(2)*S7(1))/S2
670 IF (M=0) THEN 679
671 Q2=U*U+V*V
672 X=U
673 Y=V
674 GOSUB 680
675 L1(1)=.5*LOG(Q2)
676 L1(2)=Q
677 U=L1(1)
678 V=L1(2)
679 RETURN
680 Q=ANGLE(X,Y)
681 RETURN
682 END SUB
683 OPTION NOLET
684 DIM Z(2),N(2),J(2)
685 Z(1)=7
686 Z(2)=0
687 N(1)=0
688 N(2)=0
689 CALL CBSSLJ (Z(),N(),J())
690 PRINT USING " +#.###############^^^^":Z(1),Z(2)
691 PRINT USING " +#.###############^^^^":J(1),J(2)
692 END
```

```
1 SUB QBSSLK (Z(),N,K())
2 !********************************
3 !BESSEL K OF 1/3 INTEGRAL ORDER
4 !********************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !N = ORDER (1/3 INTEGER)
7 !K = FUNCTION (COMPLEX ARRAY)
8 !
9 OPTION NOLET
10 IF (H=1) THEN 171
11 H=1
12 DIM F(5),D(60),E(60),Z1(2),L1(2)
13 DIM R1(2),S1(2),T1(2),L7(2),S7(2)
14 DIM T7(2),Q2(2),R2(2),Q7(2),F1(2)
15 DATA 0.5772156649015329E00
16 READ G
17 DATA 2.678938534707747 6E00
18 DATA 1.354117939426400 4E00
19 DATA 1.000000000000000E00
20 DATA 0.8929795115692492E00
21 DATA 0.9027452929509336E00
22 DATA 1.000000000000000E00
23 FOR I=1 TO 5
24 READ F(I)
25 NEXT I
26 DATA  0.000000000000000E00
27 DATA -1.648995051422117E-2
28 DATA -7.186218800685365E-2
29 DATA -1.670868781248656E-1
30 DATA -3.025822502194688E-1
31 DATA -4.806139452459267E-1
32 DATA -7.070752393578979E-1
33 DATA -9.929957905395160E-1
34 DATA -1.355839256125922E00
35 DATA -1.821059078991320E00
36 DATA -2.424821753108787E00
37 DATA -3.219566557087496E00
38 DATA -4.286580772483836E00
39 DATA -5.770228167981279E00
40 DATA -8.013712609525260E00
41 FOR I=1 TO 15
42 READ D(I)
43 NEXT I
44 DATA  0.000000000000000E00
45 DATA -1.436346451938738E-2
46 DATA -6.829706137621937E-2
47 DATA -1.626145943132923E-1
48 DATA -2.974538543947166E-1
49 DATA -4.749052692100660E-1
50 DATA -7.007380209350545E-1
```

```
51 DATA -9.858961485754878E-1
52 DATA -1.347791024780270E00
53 DATA -1.811850277214040E00
54 DATA -2.414232693109992E00
55 DATA -3.207380096917656E00
56 DATA -4.272584301588853E00
57 DATA -5.754199732132192E00
58 DATA -7.995346895730257E00
59 FOR I=16 TO 30
60 READ D(I)
61 NEXT I
62 DATA  0.000000000000000E00
63 DATA -1.002408161647787E-2
64 DATA -5.997870547395276E-2
65 DATA -1.516343542993466E-1
66 DATA -2.845368398965238E-1
67 DATA -4.603218236550183E-1
68 DATA -6.844133505234451E-1
69 DATA -9.675114349534103E-1
70 DATA -1.326876006320234E00
71 DATA -1.787857730177285E00
72 DATA -2.386589640411052E00
73 DATA -3.175517091435769E00
74 DATA -4.235942093386486E00
75 DATA -5.712193661037926E00
76 DATA -7.947172476979304E00
77 FOR I=31 TO 45
78 READ D(I)
79 NEXT I
80 DATA  0.000000000000000E00
81 DATA -5.577424298795054E-3
82 DATA -4.991129441724757E-2
83 DATA -1.374409116523968E-1
84 DATA -2.672337847105657E-1
85 DATA -4.403801668086817E-1
86 DATA -6.618136148725406E-1
87 DATA -9.418610776650166E-1
88 DATA -1.297541304683261E00
89 DATA -1.754076967198164E00
90 DATA -2.347552998822763E00
91 DATA -3.130413326891964E00
92 DATA -4.183971205637291E00
93 DATA -5.652517992149936E00
94 DATA -7.878639598106769E00
95 FOR I=46 TO 60
96 READ D(I)
97 NEXT I
98  DATA  0.000000000000000E00
99  DATA -4.809423363874473E-3
100 DATA -1.313662003477595E-2
```

```
101 DATA -1.948438340084579E-2
102 DATA -2.199489000320033E-2
103 DATA -2.093966256765194E-2
104 DATA -1.746002684586503E-2
105 DATA -1.279378133620848E-2
106 DATA -8.052344217965918E-3
107 DATA -4.158173750027601E-3
108 DATA -1.643177387479224E-3
109 DATA -4.491755853147087E-4
110 DATA -7.285947655740069E-5
111 DATA -5.382652306582855E-6
112 DATA -9.937790480362892E-8
113 FOR I=1 TO 15
114 READ E(I)
115 NEXT I
116 DATA  0.000000000000000E00
117 DATA -3.308130422855452E-3
118 DATA -7.992541489780344E-3
119 DATA -1.114953951149366E-2
120 DATA -1.214944767056973E-2
121 DATA -1.131050261572697E-2
122 DATA -9.290616383535283E-3
123 DATA -6.738338829864222E-3
124 DATA -4.212029053420502E-3
125 DATA -2.165634872117979E-3
126 DATA -8.537655388612388E-4
127 DATA -2.332000490440423E-4
128 DATA -3.784577154525431E-5
129 DATA -2.800396179763318E-6
130 DATA -5.183943839708712E-8
131 FOR I=16 TO 30
132 READ E(I)
133 NEXT I
134 DATA  0.000000000000000E00
135 DATA  8.356269613036070E-3
136 DATA  1.421589792080583E-2
137 DATA  1.661492960434507E-2
138 DATA  1.632135891550809E-2
139 DATA  1.421532926606566E-2
140 DATA  1.116165182456759E-2
141 DATA  7.846521235416155E-3
142 DATA  4.800776505229076E-3
143 DATA  2.433841308314365E-3
144 DATA  9.515251496981341E-4
145 DATA  2.589249109738377E-4
146 DATA  4.201905951784872E-5
147 DATA  3.118825106795326E-6
148 DATA  5.808362400385792E-8
149 FOR I=31 TO 45
150 READ E(I)
```

```
151 NEXT I
152 DATA  0.000000000000000E00
153 DATA  7.538057792005914E-2
154 DATA  7.122935374034643E-2
155 DATA  6.331162242281997E-2
156 DATA  5.282402645233011E-2
157 DATA  4.133053594414916E-2
158 DATA  3.013505739475096E-2
159 DATA  2.010434395927201E-2
160 DATA  1.185522230680744E-2
161 DATA  5.860555109560099E-3
162 DATA  2.254651482673253E-3
163 DATA  6.081730415363355E-4
164 DATA  9.842155506257467E-5
165 DATA  7.321390930380890E-6
166 DATA  1.372796673846658E-7
167 FOR I=46 TO 60
168 READ E(I)
169 NEXT I
170 RESTORE
171 Z2=Z(1)*Z(1)+Z(2)*Z(2)
172 L1(1)=.5*LOG(Z2)
173 X1=Z(1)
174 Y1=Z(2)
175 GOSUB 429
176 L1(2)=Q1
177 N0=ABS(N)
178 I2=INT(N0)
179 F2=N0-I2
180 IF (F2=0) THEN 199
181 IF (F2<.5) THEN 196
182 IF (F2>.5) THEN 198
183 F2=1.5
184 M1=SQR(PI/2)*EXP(-.5*L1(1)-Z(1))
185 F1(1)=M1*COS(-.5*L1(2)-Z(2))
186 F1(2)=M1*SIN(-.5*L1(2)-Z(2))
187 IF (N>1) THEN 191
188 S7(1)=F1(1)
189 S7(2)=F1(2)
190 GOTO 311
191 Q2(1)=F1(1)
192 Q2(2)=F1(2)
193 R2(1)=F1(1)
194 R2(2)=F1(2)
195 GOTO 298
196 F2=1
197 GOTO 199
198 F2=2
199 IF (Z2<=1) THEN 242
200 IF (Z2>=289) THEN 202
```

```
201 IF (Z(1)+.096*Z(2)*Z(2)<=0) THEN 242 ELSE 222
202 M1=SQR(PI/2)*EXP(-.5*L1(1)-Z(1))
203 F1(1)=M1*COS(-.5*L1(2)-Z(2))
204 F1(2)=M1*SIN(-.5*L1(2)-Z(2))
205 IF (N0>1) THEN 212
206 N7=I2+F2/3
207 GOSUB 314
208 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
209 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
210 S7(1)=T1(1)
211 S7(2)=T1(2)
212 GOTO 311
213 N7=1-F2/3
214 GOSUB 314
215 Q2(1)=F1(1)*S7(1)-F1(2)*S7(2)
216 Q2(2)=F1(1)*S7(2)+F1(2)*S7(1)
217 N7=F2/3
218 GOSUB 314
219 R2(1)=F1(1)*S7(1)-F1(2)*S7(2)
220 R2(2)=F1(1)*S7(2)+F1(2)*S7(1)
221 GOTO 298
222 M1=SQR(PI/2)*EXP(-.5*L1(1)-Z(1))
223 F1(1)=M1*COS(-.5*L1(2)-Z(2))
224 F1(2)=M1*SIN(-.5*L1(2)-Z(2))
225 IF (N0>1) THEN 233
226 N7=45*I2+15*F2
227 GOSUB 335
228 T1(1)=F1(1)*S7(1)-F1(2)*S7(2)
229 T1(2)=F1(1)*S7(2)+F1(2)*S7(1)
230 S7(1)=T1(1)
231 S7(2)=T1(2)
232 GOTO 311
233 N7=45-15*F2
234 GOSUB 335
235 Q2(1)=F1(1)*S7(1)-F1(2)*S7(2)
236 Q2(2)=F1(1)*S7(2)+F1(2)*S7(1)
237 N7=15*F2
238 GOSUB 335
239 R2(1)=F1(1)*S7(1)-F1(2)*S7(2)
240 R2(2)=F1(1)*S7(2)+F1(2)*S7(1)
241 GOTO 298
242 IF (F2<>0) THEN 256
243 IF (N0>1) THEN 247
244 N7=I2
245 GOSUB 349
246 GOTO 311
247 N7=1
248 GOSUB 349
249 Q2(1)=S7(1)
250 Q2(2)=S7(2)
```

```
251 N7=0
252 GOSUB 349
253 R2(1)=S7(1)
254 R2(2)=S7(2)
255 GOTO 298
256 Z7=EXP(F2*(-LOG(2)+L1(1))/3)
257 Z1(1)=Z7*COS(F2*L1(2)/3)
258 Z1(2)=Z7*SIN(F2*L1(2)/3)
259 S0=Z7*Z7
260 IF (N0>1) THEN 274
261 N7=+F2
262 Q7(1)=Z1(1)
263 Q7(2)=Z1(2)
264 GOSUB 408
265 R2(1)=S7(1)
266 R2(2)=S7(2)
267 N7=-F2
268 Q7(1)=+Z1(1)/S0
269 Q7(2)=-Z1(2)/S0
270 GOSUB 408
271 S7(1)=(PI/2)*(S7(1)-R2(1))/(SQR(3)/2)
272 S7(2)=(PI/2)*(S7(2)-R2(2))/(SQR(3)/2)
273 GOTO 311
274 N7=3-F2
275 Q7(1)=.5*(Z1(1)*Z(1)+Z1(2)*Z(2))/S0
276 Q7(2)=.5*(Z1(1)*Z(2)-Z1(2)*Z(1))/S0
277 GOSUB 408
278 Q2(1)=S7(1)
279 Q2(2)=S7(2)
280 N7=F2-3
281 Q7(1)=2*(Z(1)*Z1(1)+Z(2)*Z1(2))/Z2
282 Q7(2)=2*(Z(1)*Z1(2)-Z(2)*Z1(1))/Z2
283 GOSUB 408
284 Q2(1)=(PI/2)*(S7(1)-Q2(1))/(SQR(3)/2)
285 Q2(2)=(PI/2)*(S7(2)-Q2(2))/(SQR(3)/2)
286 N7=+F2
287 Q7(1)=Z1(1)
288 Q7(2)=Z1(2)
289 GOSUB 408
290 R2(1)=S7(1)
291 R2(2)=S7(2)
292 N7=-F2
293 Q7(1)=+Z1(1)/S0
294 Q7(2)=-Z1(2)/S0
295 GOSUB 408
296 R2(1)=(PI/2)*(S7(1)-R2(1))/(SQR(3)/2)
297 R2(2)=(PI/2)*(S7(2)-R2(2))/(SQR(3)/2)
298 R1(1)=+Z(1)/Z2
299 R1(2)=-Z(2)/Z2
300 N7=0
```

```
301 GOTO 306
302 Q2(1)=R2(1)
303 Q2(2)=R2(2)
304 R2(1)=S7(1)
305 R2(2)=S7(2)
306 N1=N7+F2/3
307 S7(1)=2*N1*(R1(1)*R2(1)-R1(2)*R2(2))+Q2(1)
308 S7(2)=2*N1*(R1(1)*R2(2)+R1(2)*R2(1))+Q2(2)
309 N7=N7+1
310 IF (N7<I2) THEN 302
311 K(1)=S7(1)
312 K(2)=S7(2)
313 EXIT SUB
314 S7(1)=0
315 S7(2)=0
316 R1(1)=+.5*Z(1)/Z2
317 R1(2)=-.5*Z(2)/Z2
318 N1=(N7-.5)*(N7+.5)
319 T7(1)=1
320 T7(2)=0
321 M7=0
322 GOTO 331
323 N1=N1-2*M7
324 M7=M7+1
325 T1(1)=R1(1)*T7(1)-R1(2)*T7(2)
326 T1(2)=R1(1)*T7(2)+R1(2)*T7(1)
327 T7(1)=N1*T1(1)/M7
328 T7(2)=N1*T1(2)/M7
329 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 331
330 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 334
331 S7(1)=S7(1)+T7(1)
332 S7(2)=S7(2)+T7(2)
333 IF (M7<36) THEN 323
334 RETURN
335 S7(1)=1
336 S7(2)=0
337 J=N7+2
338 L=N7+15
339 FOR I=J TO L
340 T1(1)=Z(1)-D(I)
341 T1(2)=Z(2)
342 S2=T1(1)*T1(1)+T1(2)*T1(2)
343 T7(1)=+E(I)*T1(1)/S2
344 T7(2)=-E(I)*T1(2)/S2
345 S7(1)=S7(1)+T7(1)
346 S7(2)=S7(2)+T7(2)
347 NEXT I
348 RETURN
349 Q7(1)=1
350 Q7(2)=0
```

```
351 N2=0
352 S0=-1
353 M7=0
354 GOTO 362
355 M7=M7+1
356 N2=N2+.5/M7
357 S0=-S0
358 T1(1)=.5*(Z(1)*Q7(1)-Z(2)*Q7(2))
359 T1(2)=.5*(Z(1)*Q7(2)+Z(2)*Q7(1))
360 Q7(1)=T1(1)/M7
361 Q7(2)=T1(2)/M7
362 IF (M7<N7) THEN 355
363 S1(1)=.25*(Z(1)-Z(2))*(Z(1)+Z(2))
364 S1(2)=.5*Z(1)*Z(2)
365 R2(1)=0
366 R2(2)=0
367 S2=Q7(1)*Q7(1)+Q7(2)*Q7(2)
368 T7(1)=+Q7(1)/S2
369 T7(2)=-Q7(2)/S2
370 M7=0
371 GOTO 381
372 T7(1)=T7(1)/(N7-M7)
373 T7(2)=T7(2)/(N7-M7)
374 R2(1)=R2(1)+.5*T7(1)
375 R2(2)=R2(2)+.5*T7(2)
376 M7=M7+1
377 T1(1)=S1(1)*T7(1)-S1(2)*T7(2)
378 T1(2)=S1(1)*T7(2)+S1(2)*T7(1)
379 T7(1)=-T1(1)/M7
380 T7(2)=-T1(2)/M7
381 IF (M7<N7) THEN 372
382 S7(1)=0
383 S7(2)=0
384 M2=1
385 M1=0
386 Q7(1)=S0*Q7(1)
387 Q7(2)=S0*Q7(2)
388 L7(1)=-LOG(2)+G+L1(1)-N2
389 L7(2)=+L1(2)
390 M7=0
391 GOTO 400
392 M1=M1+M2
393 M7=M7+1
394 M2=.25*Z2*M2/(M7*(N7+M7))
395 T1(1)=S1(1)*Q7(1)-S1(2)*Q7(2)
396 T1(2)=S1(1)*Q7(2)+S1(2)*Q7(1)
397 Q7(1)=T1(1)/(M7*(N7+M7))
398 Q7(2)=T1(2)/(M7*(N7+M7))
399 L7(1)=L7(1)-.5/M7-.5/(N7+M7)
400 T7(1)=Q7(1)*L7(1)-Q7(2)*L7(2)
```

```
401 T7(2)=Q7(1)*L7(2)+Q7(2)*L7(1)
402 S7(1)=S7(1)+T7(1)
403 S7(2)=S7(2)+T7(2)
404 IF (M1+M2>M1) THEN 392
405 S7(1)=R2(1)+S7(1)
406 S7(2)=R2(2)+S7(2)
407 RETURN
408 S7(1)=0
409 S7(2)=0
410 S1(1)=.25*(Z(1)-Z(2))*(Z(1)+Z(2))
411 S1(2)=.5*Z(1)*Z(2)
412 N0=3+N7
413 T7(1)=Q7(1)/F(N0)
414 T7(2)=Q7(2)/F(N0)
415 N7=N7/3
416 M7=0
417 GOTO 425
418 M7=M7+1
419 T1(1)=S1(1)*T7(1)-S1(2)*T7(2)
420 T1(2)=S1(1)*T7(2)+S1(2)*T7(1)
421 T7(1)=T1(1)/(M7*(N7+M7))
422 T7(2)=T1(2)/(M7*(N7+M7))
423 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 425
424 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 428
425 S7(1)=S7(1)+T7(1)
426 S7(2)=S7(2)+T7(2)
427 GOTO 418
428 RETURN
429 Q1=ANGLE(X1,Y1)
430 RETURN
431 END SUB
432 OPTION NOLET
433 DIM Z(2),K(2)
434 Z(1)=1
435 Z(2)=0
436 N=2/3
437 CALL QBSSLK (Z(),N,K())
438 PRINT USING " +#.###############^^^^": K(1),K(2)
439 PRINT USING " +#.###############^^^^": EXP(Z(1))*K(1)
440 END
```

```
1 SUB CGAMMA (M,Z(),G())
2 !******************************
3 !SUBROUTINE FOR GAMMA FUNCTION
4 !******************************
5 !M = MODE OF OPERATION
6 !Z = ARGUMENT (COMPLEX ARRAY)
7 !G = FUNCTION (COMPLEX ARRAY)
8 OPTION NOLET
9 IF (H=1) THEN 65
10 H=1
11 DIM C(26),B(18)
12 DIM F1(2),L1(2),L2(2),L3(2)
13 DIM R1(2),S1(2),S7(2),T1(2),T7(2)
14 DATA +1.0000000000000000E0
15 DATA +0.5772156649015329E0
16 DATA -0.6558780715202538E0
17 DATA -0.0420026350340952E0
18 DATA +0.1665386113822915E0
19 DATA -0.0421977345555443E0
20 DATA -0.0096219715278770E0
21 DATA +0.0072189432466630E0
22 DATA -0.0011651675918591E0
23 DATA -0.0002152416741149E0
24 DATA +0.0001280502823882E0
25 DATA -0.0000201348547807E0
26 DATA -0.0000012504934821E0
27 DATA +0.0000011330272320E0
28 DATA -0.0000002056338417E0
29 DATA +0.0000000061160950E0
30 DATA +0.0000000050020075E0
31 DATA -0.0000000011812746E0
32 DATA +0.0000000001043427E0
33 DATA +0.0000000000077823E0
34 DATA -0.0000000000036968E0
35 DATA +0.0000000000005100E0
36 DATA -0.0000000000000206E0
37 DATA -0.0000000000000054E0
38 DATA +0.0000000000000014E0
39 DATA +0.0000000000000001E0
40 FOR I=1 TO 26
41 READ C(I)
42 NEXT I
43 DATA +8.333333333333333E-2
44 DATA -2.777777777777777E-3
45 DATA +7.936507936507937E-4
46 DATA -5.952380952380952E-4
47 DATA +8.417508417508418E-4
48 DATA -1.917526917526918E-3
49 DATA +6.410256410256410E-3
50 DATA -2.955065359477124E-2
```

```
51 DATA +1.796443723688306E-1
52 DATA -1.392432216905901E00
53 DATA +1.340286404416839E+1
54 DATA -1.568482846260020E+2
55 DATA +2.193103333333333E+3
56 DATA -3.610877125372499E+4
57 DATA +6.914722688513131E+5
58 DATA -1.523822153940742E+7
59 DATA +3.829007513914141E+8
60 DATA -1.088226603578439E10
61 FOR I=1 TO 18
62 READ B(I)
63 NEXT I
64 RESTORE
65 P1=IP(ABS(Z(1))+.5)
66 P1=ABS(P1)*SGN(Z(1))
67 IF (P1>0) THEN 101
68 X1=PI*(Z(1)-P1)
69 Y1=PI*Z(2)
70 S1(1)=SIN(X1)*.5*(EXP(+Y1)+EXP(-Y1))
71 S1(2)=COS(X1)*.5*(EXP(+Y1)-EXP(-Y1))
72 X1=1-Z(1)
73 Y1=-Z(2)
74 Q1=1-P1
75 GOSUB 108
76 IF (M<>0) THEN 88
77 N7=P1
78 Q1=2*IP(N7/2)
79 IF (P1=Q1) THEN 82
80 S1(1)=-S1(1)
81 S1(2)=-S1(2)
82 T1(1)=S1(1)*U1-S1(2)*V1
83 T1(2)=S1(1)*V1+S1(2)*U1
84 S2=T1(1)*T1(1)+T1(2)*T1(2)
85 G(1)=+PI*T1(1)/S2
86 G(2)=-PI*T1(2)/S2
87 EXIT SUB
88 S2=S1(1)*S1(1)+S1(2)*S1(2)
89 X=S1(1)
90 Y=S1(2)
91 GOSUB 195
92 L2(1)=.5*LOG(S2)
93 L2(2)=Q
94 IF (Z(2)>=0) THEN 97
95 L2(2)=L2(2)+PI*P1
96 GOTO 98
97 L2(2)=L2(2)-PI*P1
98 G(1)=LOG(PI)-L2(1)-U1
99 G(2)=-L2(2)-V1
100 EXIT SUB
```

```
101 X1=Z(1)
102 Y1=Z(2)
103 Q1=P1
104 GOSUB 108
105 G(1)=U1
106 G(2)=V1
107 EXIT SUB
108 L1(1)=0
109 L1(2)=0
110 S2=X1*X1+Y1*Y1
111 IF (S2>=32) THEN 126
112 F1(1)=1
113 F1(2)=0
114 X2=X1-Q1
115 Y2=Y1
116 T2=X2*X2+Y2*Y2
117 IF (T2<=1) THEN 164
118 X=X1
119 Y=Y1
120 GOSUB 195
121 L1(1)=L1(1)+.5*LOG(S2)
122 L1(2)=L1(2)+Q
123 X1=X1+1
124 S2=X1*X1+Y1*Y1
125 IF (S2<32) THEN 118
126 X=X1
127 Y=Y1
128 GOSUB 195
129 L3(1)=.5*LOG(S2)
130 L3(2)=Q
131 R1(1)=(X1-.5)*L3(1)-Y1*L3(2)-X1+.5*LOG(2*PI)
132 R1(2)=(X1-.5)*L3(2)+Y1*L3(1)-Y1
133 T7(1)=+X1/S2
134 T7(2)=-Y1/S2
135 F1(1)=T7(1)*T7(1)-T7(2)*T7(2)
136 F1(2)=2*T7(1)*T7(2)
137 S7(1)=0
138 S7(2)=0
139 FOR M7=1 TO 18
140 T1(1)=B(M7)*T7(1)
141 T1(2)=B(M7)*T7(2)
142 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 144
143 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 151
144 S7(1)=S7(1)+T1(1)
145 S7(2)=S7(2)+T1(2)
146 T1(1)=T7(1)*F1(1)-T7(2)*F1(2)
147 T1(2)=T7(1)*F1(2)+T7(2)*F1(1)
148 T7(1)=T1(1)
149 T7(2)=T1(2)
150 NEXT M7
```

```
151 U1=R1(1)+S7(1)-L1(1)
152 V1=R1(2)+S7(2)-L1(2)
153 IF (M<>0) THEN 159
154 Q2=EXP(U1)
155 F1(1)=Q2*COS(V1)
156 F1(2)=Q2*SIN(V1)
157 U1=F1(1)
158 V1=F1(2)
159 RETURN
160 T1(1)=F1(1)*X1-F1(2)*Y1
161 T1(2)=F1(1)*Y1+F1(2)*X1
162 F1(1)=T1(1)
163 F1(2)=T1(2)
164 X1=X1-1
165 IF (X1>X2) THEN 160
166 T7(1)=1
167 T7(2)=0
168 S7(1)=0
169 S7(2)=0
170 FOR M7=1 TO 26
171 T1(1)=C(M7)*T7(1)
172 T1(2)=C(M7)*T7(2)
173 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 175
174 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 182
175 S7(1)=S7(1)+T1(1)
176 S7(2)=S7(2)+T1(2)
177 T1(1)=X1*T7(1)-Y1*T7(2)
178 T1(2)=X1*T7(2)+Y1*T7(1)
179 T7(1)=T1(1)
180 T7(2)=T1(2)
181 NEXT M7
182 S2=S7(1)*S7(1)+S7(2)*S7(2)
183 U1=+(F1(1)*S7(1)+F1(2)*S7(2))/S2
184 V1=-(F1(1)*S7(2)-F1(2)*S7(1))/S2
185 IF (M=0) THEN 194
186 Q2=U1*U1+V1*V1
187 X=U1
188 Y=V1
189 GOSUB 195
190 L1(1)=.5*LOG(Q2)
191 L1(2)=Q
192 U1=L1(1)
193 V1=L1(2)
194 RETURN
195 Q=ANGLE(X,Y)
196 RETURN
197 END SUB
198 OPTION NOLET
199 DIM Z(2),G(2),L(2)
200 Z(1)=2
```

```
201 Z(2)=7
202 M=0
203 CALL CGAMMA (M,Z(),G())
204 PRINT USING "+#.##########^^^^": G(1)
205 PRINT USING "+#.##########^^^^": G(2)
206 S=G(1)*G(1)+G(2)*G(2)
207 L(1)=.5*LOG(S)
208 L(2)=2*PI+ANGLE(G(1),G(2))
209 PRINT USING "+#.##########^^^^": L(1)
210 PRINT USING "+#.##########^^^^": L(2)
211 M=1
212 CALL CGAMMA (M,Z(),G())
213 PRINT USING "+#.##########^^^^": G(1)
214 PRINT USING "+#.##########^^^^": G(2)
215 END
```

```
1 SUB EXPLI (M,Z(),F())
2 !***********************************
3 !SUBROUTINE FOR EXPONENTIAL INTEGRAL
4 !***********************************
5 ! Z = ARGUMENT (COMPLEX ARRAY)
6 ! F = INTEGRAL (COMPLEX ARRAY)
7 OPTION NOLET
8 IF (H=1) THEN 57
9 H=1
10 DIM D(18),E(18),L1(2),T1(2)
11 DIM Q2(2),R1(2),S7(2),T7(2)
12 DATA 0.5772156649015329E00
13 READ G
14 DATA 0.000000000000000E00
15 DATA 3.111059570865283E-2
16 DATA 1.036612605391116E-1
17 DATA 2.165323352445536E-1
18 DATA 3.699314279601916E-1
19 DATA 5.667662599905892E-1
20 DATA 8.140420663247483E-1
21 DATA 1.123842475408128E00
22 DATA 1.514004781485123E00
23 DATA 2.008867950322836E00
24 DATA 2.640524118235915E00
25 DATA 3.450984499333923E00
26 DATA 4.495833607632020E00
27 DATA 5.850582634098224E00
28 DATA 7.622735014633804E00
29 DATA 9.978145015845783E00
30 DATA 1.321220648964083E+1
31 DATA 1.803229483760214E+1
32 FOR I=1 TO 18
33 READ D(I)
34 NEXT I
35 DATA 8.501565161210931E-3
36 DATA 5.050374658490585E-2
37 DATA 8.368173689564071E-2
38 DATA 1.070475824176067E-1
39 DATA 1.204247190294619E-1
40 DATA 1.250966315822293E-1
41 DATA 1.223144352246853E-1
42 DATA 1.126214175539071E-1
43 DATA 9.634194073925819E-2
44 DATA 7.473984227575106E-2
45 DATA 5.085961359534413E-2
46 DATA 2.908227067736279E-2
47 DATA 1.322016405301009E-2
48 DATA 4.438029398290675E-3
49 DATA 9.926124789875764E-4
50 DATA 1.265797951120111E-4
```

```
51 DATA 7.021509082533505E-6
52 DATA 9.102815325646319E-8
53 FOR I=1 TO 18
54 READ E(I)
55 NEXT I
56 RESTORE
57 Z2=Z(1)*Z(1)+Z(2)*Z(2)
58 IF (Z2<=1) THEN 92
59 IF (Z2>=1600) THEN 61
60 IF (-Z(1)+.064*Z(2)*Z(2)>0) THEN 80 ELSE 92
61 S7(1)=0
62 S7(2)=0
63 Q2(1)=+Z(1)/Z2
64 Q2(2)=-Z(2)/Z2
65 T7(1)=Q2(1)
66 T7(2)=Q2(2)
67 M7=0
68 GOTO 76
69 M7=M7+1
70 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
71 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
72 T7(1)=M7*T1(1)
73 T7(2)=M7*T1(2)
74 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 76
75 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 119
76 S7(1)=S7(1)+T7(1)
77 S7(2)=S7(2)+T7(2)
78 IF (M7<40) THEN 69
79 GOTO 119
80 S7(1)=0
81 S7(2)=0
82 FOR I=1 TO 18
83 T1(1)=Z(1)-D(I)
84 T1(2)=Z(2)
85 S2=T1(1)*T1(1)+T1(2)*T1(2)
86 T7(1)=+E(I)*T1(1)/S2
87 T7(2)=-E(I)*T1(2)/S2
88 S7(1)=S7(1)+T7(1)
89 S7(2)=S7(2)+T7(2)
90 NEXT I
91 GOTO 119
92 L1(1)=.5*LOG(Z2)
93 X7=-Z(1)
94 Y7=-Z(2)
95 GOSUB 135
96 L1(2)=Q7
97 R1(1)=G+L1(1)+Z(1)
98 R1(2)=L1(2)+Z(2)
99  S7(1)=0
100 S7(2)=0
```

```
101 T1(1)=Z(1)
102 T1(2)=Z(2)
103 M7=1
104 M7=M7+1
105 T7(1)=T1(1)*Z(1)-T1(2)*Z(2)
106 T7(2)=T1(1)*Z(2)+T1(2)*Z(1)
107 T1(1)=T7(1)/M7
108 T1(2)=T7(2)/M7
109 T7(1)=T1(1)/M7
110 T7(2)=T1(2)/M7
111 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 113
112 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 116
113 S7(1)=S7(1)+T7(1)
114 S7(2)=S7(2)+T7(2)
115 GOTO 104
116 S7(1)=S7(1)+R1(1)
117 S7(2)=S7(2)+R1(2)
118 GOTO 124
119 IF (M<>0) THEN 132
120 Q1=EXP(+Z(1))
121 Q2(1)=Q1*COS(Z(2))
122 Q2(2)=Q1*SIN(Z(2))
123 GOTO 128
124 IF (M=0) THEN 132
125 Q1=EXP(-Z(1))
126 Q2(1)=Q1*COS(-Z(2))
127 Q2(2)=Q1*SIN(-Z(2))
128 T1(1)=Q2(1)*S7(1)-Q2(2)*S7(2)
129 T1(2)=Q2(1)*S7(2)+Q2(2)*S7(1)
130 S7(1)=T1(1)
131 S7(2)=T1(2)
132 F(1)=S7(1)
133 F(2)=S7(2)
134 EXIT SUB
135 Q7=ANGLE(X7,Y7)
136 RETURN
137 END SUB
138 OPTION NOLET
139 DIM Z(2),F(2)
140 Z(1)=7
141 Z(2)=0
142 M=0
143 CALL EXPLI (M,Z(),F())
144 PRINT USING "+#.##########^^^^": F(1)
145 PRINT USING "+#.##########^^^^": F(2)
146 M=1
147 CALL EXPLI (M,Z(),F())
148 PRINT USING "+#.##########^^^^": F(1)
149 PRINT USING "+#.##########^^^^": F(2)
150 EX=EXP(Z(1))
```

EXPLI

```
151 PRINT USING "+#.##########^^^^": EX*F(1)
152 PRINT USING "+#.##########^^^^": EX*F(2)
153 END
```

```
1 SUB XPNLI (M, Z(), F())
2 !****************************
3 !SPECIAL EXPONENTIAL INTEGRAL
4 !****************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !F = INTEGRAL (COMPLEX ARRAY)
7 OPTION NOLET
8 IF (H=1) THEN 57
9 H=1
10 DIM D(18),E(18),L1(2),T1(2)
11 DIM Q2(2),R1(2),S7(2),T7(2)
12 DATA 0.5772156649015329E00
13 READ G
14 DATA 0.000000000000000E00
15 DATA 3.111059570865283E-2
16 DATA 1.036612605391116E-1
17 DATA 2.165323352445536E-1
18 DATA 3.699314279601916E-1
19 DATA 5.667662599905892E-1
20 DATA 8.140420663247483E-1
21 DATA 1.123842475408128E00
22 DATA 1.514004781485123E00
23 DATA 2.008867950322836E00
24 DATA 2.640524118235915E00
25 DATA 3.450984499333923E00
26 DATA 4.495833607632020E00
27 DATA 5.850582634098224E00
28 DATA 7.622735014633804E00
29 DATA 9.978145015845783E00
30 DATA 1.321220648964083E+1
31 DATA 1.803229483760214E+1
32 FOR I=1 TO 18
33 READ D(I)
34 NEXT I
35 DATA-9.914984348387891E-1
36 DATA 5.050374658490585E-2
37 DATA 8.368173689564071E-2
38 DATA 1.070475824176067E-1
39 DATA 1.204247190294619E-1
40 DATA 1.250966315822293E-1
41 DATA 1.223144352246853E-1
42 DATA 1.126214175539071E-1
43 DATA 9.634194073925819E-2
44 DATA 7.473984227575106E-2
45 DATA 5.085961359534413E-2
46 DATA 2.908227067736279E-2
47 DATA 1.322016405301009E-2
48 DATA 4.438029398290675E-3
49 DATA 9.926124789875764E-4
50 DATA 1.265797951120111E-4
```

```
51 DATA 7.021509082533505E-6
52 DATA 9.102815325646319E-8
53 FOR I=1 TO 18
54 READ E(I)
55 NEXT I
56 RESTORE
57 Z2=Z(1)*Z(1)+Z(2)*Z(2)
58 IF (Z2<=1) THEN 92
59 IF (Z2>=1600) THEN 62
60 IF (-Z(1)+.064*Z(2)*Z(2)>0) THEN 80
61 GOTO 92
62 S7(1)=0
63 S7(2)=0
64 Q2(1)=+Z(1)/Z2
65 Q2(2)=-Z(2)/Z2
66 T7(1)=Q2(1)
67 T7(2)=Q2(2)
68 M7=0
69 M7=M7+1
70 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
71 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
72 T7(1)=M7*T1(1)
73 T7(2)=M7*T1(2)
74 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 76
75 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 119
76 S7(1)=S7(1)+T7(1)
77 S7(2)=S7(2)+T7(2)
78 IF (M7<40) THEN 69
79 GOTO 119
80 S7(1)=0
81 S7(2)=0
82 FOR I=1 TO 18
83 T1(1)=Z(1)-D(I)
84 T1(2)=Z(2)
85 S2=T1(1)*T1(1)+T1(2)*T1(2)
86 T7(1)=+E(I)*T1(1)/S2
87 T7(2)=-E(I)*T1(2)/S2
88 S7(1)=S7(1)+T7(1)
89 S7(2)=S7(2)+T7(2)
90 NEXT I
91 GOTO 119
92 L1(1)=.5*LOG(Z2)
93 X7=-Z(1)
94 Y7=-Z(2)
95 GOSUB 146
96 L1(2)=Q7
97 R1(1)=G+L1(1)+Z(1)
98 R1(2)=L1(2)+Z(2)
99  S7(1)=0
100 S7(2)=0
```

```
101 T7(1)=Z(1)
102 T7(2)=Z(2)
103 M7=1
104 M7=M7+1
105 T1(1)=T7(1)*Z(1)-T7(2)*Z(2)
106 T1(2)=T7(1)*Z(2)+T7(2)*Z(1)
107 T7(1)=T1(1)/M7
108 T7(2)=T1(2)/M7
109 T1(1)=T7(1)/M7
110 T1(2)=T7(2)/M7
111 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 113
112 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 116
113 S7(1)=S7(1)+T1(1)
114 S7(2)=S7(2)+T1(2)
115 GOTO 104
116 S7(1)=S7(1)+R1(1)
117 S7(2)=S7(2)+R1(2)
118 GOTO 128
119 IF (M<>0) THEN 143
120 Q1=EXP(Z(1))
121 Q2(1)=Q1*COS(Z(2))
122 Q2(2)=Q1*SIN(Z(2))
123 T1(1)=Q2(1)*S7(1)-Q2(2)*S7(2)
124 T1(2)=Q2(1)*S7(2)+Q2(2)*S7(1)
125 S7(1)=T1(1)
126 S7(2)=T1(2)
127 GOTO 143
128 Q1=EXP(-Z(1))
129 T7(1)=COS(-Z(2))
130 T7(2)=SIN(-Z(2))
131 Q2(1)=+Z(1)/Z2
132 Q2(2)=-Z(2)/Z2
133 IF (M<>0) THEN 139
134 T1(1)=(T7(1)*Q2(1)+T7(2)*Q2(2))/Q1
135 T1(2)=(T7(1)*Q2(2)-T7(2)*Q2(1))/Q1
136 S7(1)=S7(1)-T1(1)
137 S7(2)=S7(2)-T1(2)
138 GOTO 143
139 T1(1)=(T7(1)*S7(1)-T7(2)*S7(2))*Q1
140 T1(2)=(T7(1)*S7(2)+T7(2)*S7(1))*Q1
141 S7(1)=T1(1)-Q2(1)
142 S7(2)=T1(2)-Q2(2)
143 F(1)=S7(1)
144 F(2)=S7(2)
145 EXIT SUB
146 Q7=ANGLE(X7,Y7)
147 RETURN
148 END SUB
149 OPTION NOLET
150 DIM Z(2),F(2),X(2),Q(2)
```

XPNLI

```
151 Z(1)=7
152 Z(2)=0
153 M=0
154 CALL XPNLI (M,Z(),X())
155 Z2=Z(1)*Z(1)+Z(2)*Z(2)
156 F1=EXP(Z(1))
157 F(1)=F1*COS(Z(2))
158 F(2)=F1*SIN(Z(2))
159 Q(1)=(Z(1)*F(1)+Z(2)*F(2))/Z2
160 Q(2)=(Z(1)*F(2)-Z(2)*F(1))/Z2
161 F(1)=X(1)+Q(1)
162 F(2)=X(2)+Q(2)
163 PRINT USING " +#.###############^^^^": X(1),X(2)
164 PRINT USING " +#.###############^^^^": F(1),F(2)
165 END
```

```
1 SUB FRNLI (M,Z(),F()) STATIC
2 !*******************************
3 !SUBROUTINE FOR FRESNEL INTEGRAL
4 !*******************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !F = INTEGRAL (COMPLEX ARRAY)
7 OPTION NOLET
8 IF (H=1) THEN 57
9 H=1
10 DIM D(18),E(18),Q2(2),T1(2)
11 DIM R1(2),R2(2),S7(2),T7(2)
12 DATA 0.564189583547756E00
13 READ R
14 DATA 0.000000000000000E00
15 DATA 2.086058560134765E-2
16 DATA 8.298069404956873E-2
17 DATA 1.854216533260787E-1
18 DATA 3.279634793823607E-1
19 DATA 5.126752799128284E-1
20 DATA 7.454129580451047E-1
21 DATA 1.036950674182965E00
22 DATA 1.403780612554370E00
23 DATA 1.868916622140010E00
24 DATA 2.463148305239293E00
25 DATA 3.227193837373523E00
26 DATA 4.215343482800130E00
27 DATA 5.501788731515490E00
28 DATA 7.192589666831019E00
29 DATA 9.451702080764080E00
30 DATA 1.257107183147839E+1
31 DATA 1.724835372163339E+1
32 FOR I=1 TO 18
33 READ D(I)
34 NEXT I
35 DATA 8.157230833240962E-2
36 DATA 1.592852852534368E-1
37 DATA 1.485816256144991E-1
38 DATA 1.332196708362453E-1
39 DATA 1.156903928789572E-1
40 DATA 9.785809594475354E-2
41 DATA 8.059088342976243E-2
42 DATA 6.402045386098722E-2
43 DATA 4.814452427678847E-2
44 DATA 3.335406584732945E-2
45 DATA 2.055480994701934E-2
46 DATA 1.078474038875057E-2
47 DATA 4.556348922142192E-3
48 DATA 1.439844581389254E-3
49 DATA 3.070561398341705E-4
50 DATA 3.781565411685414E-5
```

```
51 DATA 2.051735096161211E-6
52 DATA 2.635648236827474E-8
53 FOR I=1 TO 18
54 READ E(I)
55 NEXT I
56 RESTORE
57 Z2=Z(1)*Z(1)+Z(2)*Z(2)
58 Z1=SQR(Z2)
59 R2(1)=0
60 R2(2)=0
61 IF (Z(1)=0) THEN 67
62 IF (Z(1)>0) THEN 68
63 R2(2)=SQR(-Z(1)+Z1)
64 R2(1)=Z(2)/R2(2)
65 Q1=+R
66 GOTO 72
67 IF (Z(2)=0) THEN 74
68 R2(1)=SQR(+Z(1)+Z1)
69 R2(2)=Z(2)/R2(1)
70 IF (Z(2)>0) THEN 65
71 Q1=-R
72 R2(1)=Q1*R2(1)
73 R2(2)=Q1*R2(2)
74 IF (Z2<=1) THEN 108
75 IF (Z2>=1444) THEN 77
76 IF (-Z(1)+.064*Z(2)*Z(2)>0) THEN 96 ELSE 108
77 S7(1)=0
78 S7(2)=0
79 Q2(1)=+Z(1)/Z2
80 Q2(2)=-Z(2)/Z2
81 T7(1)=Q2(1)
82 T7(2)=Q2(2)
83 M7=-.5
84 GOTO 92
85 M7=M7+1
86 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
87 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
88 T7(1)=M7*T1(1)
89 T7(2)=M7*T1(2)
90 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 92
91 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 131
92 S7(1)=S7(1)+T7(1)
93 S7(2)=S7(2)+T7(2)
94 IF (M7<37.5) THEN 85
95 GOTO 131
96 S7(1)=0
97 S7(2)=0
98 FOR I=1 TO 18
99  T1(1)=Z(1)-D(I)
100 T1(2)=Z(2)
```

```
101 S2=T1(1)*T1(1)+T1(2)*T1(2)
102 T7(1)=+E(I)*T1(1)/S2
103 T7(2)=-E(I)*T1(2)/S2
104 S7(1)=S7(1)+T7(1)
105 S7(2)=S7(2)+T7(2)
106 NEXT I
107 GOTO 131
108 R1(1)=R2(1)
109 R1(2)=-SQR(.5)+R2(2)
110 S7(1)=0
111 S7(2)=0
112 T7(1)=R2(1)
113 T7(2)=R2(2)
114 M7=0
115 M7=M7+1
116 N7=2*M7+1
117 T1(1)=T7(1)*Z(1)-T7(2)*Z(2)
118 T1(2)=T7(1)*Z(2)+T7(2)*Z(1)
119 T7(1)=T1(1)/M7
120 T7(2)=T1(2)/M7
121 T1(1)=T7(1)/N7
122 T1(2)=T7(2)/N7
123 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 125
124 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 128
125 S7(1)=S7(1)+T1(1)
126 S7(2)=S7(2)+T1(2)
127 GOTO 115
128 S7(1)=S7(1)+R1(1)
129 S7(2)=S7(2)+R1(2)
130 GOTO 140
131 T1(1)=R2(1)*S7(1)-R2(2)*S7(2)
132 T1(2)=R2(1)*S7(2)+R2(2)*S7(1)
133 S7(1)=.5*T1(1)
134 S7(2)=.5*T1(2)
135 IF (M<>0) THEN 148
136 Q1=EXP(+Z(1))
137 Q2(1)=Q1*COS(+Z(2))
138 Q2(2)=Q1*SIN(+Z(2))
139 GOTO 144
140 IF (M=0) THEN 148
141 Q1=EXP(-Z(1))
142 Q2(1)=Q1*COS(-Z(2))
143 Q2(2)=Q1*SIN(-Z(2))
144 T1(1)=Q2(1)*S7(1)-Q2(2)*S7(2)
145 T1(2)=Q2(1)*S7(2)+Q2(2)*S7(1)
146 S7(1)=T1(1)
147 S7(2)=T1(2)
148 F(1)=S7(1)
149 F(2)=S7(2)
150 EXIT SUB
```

FRNLI

```
151 END SUB
152 OPTION NOLET
153 DIM Z(2),F(2)
154 Z(1)=0
155 Z(2)=PI/2
156 M=0
157 CALL FRNLI (M,Z(),F())
158 C=.5+SQR(.5)*(F(1)+F(2))
159 S=.5-SQR(.5)*(F(1)-F(2))
160 PRINT USING "+#.##########^^^^": C
161 PRINT USING "+#.##########^^^^": S
162 PRINT USING "+#.##########^^^^": F(1)
163 PRINT USING "+#.##########^^^^": F(2)
164 M=1
165 CALL FRNLI (M,Z(),F())
166 PRINT USING "+#.##########^^^^": -F(2)
167 PRINT USING "+#.##########^^^^": +F(1)
168 END
```

```
1 SUB CERF (M,Z(),F())
2 !******************************
3 !SUBROUTINE FOR ERROR FUNCTION
4 !******************************
5 !Z = ARGUMENT (COMPLEX ARRAY)
6 !F = FUNCTION (COMPLEX ARRAY)
7 OPTION NOLET
8 IF (H=1) THEN 57
9 H=1
10 DIM D(18),E(18),Q2(2),T1(2)
11 DIM R1(2),R2(2),S7(2),T7(2)
12 DATA 0.564189583547756E00
13 READ R
14 DATA 0.000000000000000E00
15 DATA 2.086058560134765E-2
16 DATA 8.298069404956873E-2
17 DATA 1.854216533260787E-1
18 DATA 3.279634793823607E-1
19 DATA 5.126752799128284E-1
20 DATA 7.454129580451047E-1
21 DATA 1.036950674182965E00
22 DATA 1.403780612554370E00
23 DATA 1.868916622140010E00
24 DATA 2.463148305239293E00
25 DATA 3.227193837373523E00
26 DATA 4.215343482800130E00
27 DATA 5.501788731515490E00
28 DATA 7.192589666831019E00
29 DATA 9.451702080764080E00
30 DATA 1.257107183147839E+1
31 DATA 1.724835372163339E+1
32 FOR I=1 TO 18
33 READ D(I)
34 NEXT I
35 DATA 8.157230833240962E-2
36 DATA 1.592852852534368E-1
37 DATA 1.485816256144991E-1
38 DATA 1.332196708362453E-1
39 DATA 1.156903928789572E-1
40 DATA 9.785809594475354E-2
41 DATA 8.059088342976243E-2
42 DATA 6.402045386098722E-2
43 DATA 4.814452427678847E-2
44 DATA 3.335406584732945E-2
45 DATA 2.055480994701934E-2
46 DATA 1.078474038875057E-2
47 DATA 4.556348922142192E-3
48 DATA 1.439844581389254E-3
49 DATA 3.070561398341705E-4
50 DATA 3.781565411685414E-5
```

```
51 DATA 2.05173509616121lE-6
52 DATA 2.635648236827474E-8
53 FOR I=1 TO 18
54 READ E(I)
55 NEXT I
56 RESTORE
57 Z2=Z(1)*Z(1)+Z(2)*Z(2)
58 R2(1)=Z(1)*Z(1)-Z(2)*Z(2)
59 R2(2)=2*Z(1)*Z(2)
60 R1(1)=+Z(1)
61 R1(2)=+Z(2)
62 S0=+1
63 IF (Z(1)>=0) THEN 67
64 R1(1)=-Z(1)
65 R1(2)=-Z(2)
66 S0=-1
67 IF (Z2<=1) THEN 125
68 IF (Z2>=38) THEN 70
69 IF (R2(1)+.064*R2(2)*R2(2)>0) THEN 100 ELSE 125
70 S7(1)=0
71 S7(2)=0
72 Q2(1)=+R2(1)/(Z2*Z2)
73 Q2(2)=-R2(2)/(Z2*Z2)
74 Q1=R*EXP(-R2(1))
75 T1(1)=Q1*COS(-R2(2))
76 T1(2)=Q1*SIN(-R2(2))
77 T7(1)=+(T1(1)*R1(1)+T1(2)*R1(2))/Z2
78 T7(2)=-(T1(1)*R1(2)-T1(2)*R1(1))/Z2
79 M7=-.5
80 GOTO 88
81 M7=M7+1
82 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
83 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
84 T7(1)=-M7*T1(1)
85 T7(2)=-M7*T1(2)
86 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 88
87 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 91
88 S7(1)=S7(1)+T7(1)
89 S7(2)=S7(2)+T7(2)
90 IF (M7<37.5) THEN 81
91 F(1)=S0*S7(1)
92 F(2)=S0*S7(2)
93 IF (M=0) THEN 97
94 IF (S0=1) THEN 96
95 F(1)=2+F(1)
96 EXIT SUB
97 F(1)=S0-F(1)
98 F(2)=-F(2)
99 EXIT SUB
100 S7(1)=0
```

```
101 S7(2)=0
102 Q1=R*EXP(-R2(1))
103 T1(1)=Q1*COS(-R2(2))
104 T1(2)=Q1*SIN(-R2(2))
105 Q2(1)=T1(1)*R1(1)-T1(2)*R1(2)
106 Q2(2)=T1(1)*R1(2)+T1(2)*R1(1)
107 FOR I=1 TO 18
108 T1(1)=R2(1)+D(I)
109 T1(2)=R2(2)
110 S2=T1(1)*T1(1)+T1(2)*T1(2)
111 T7(1)=+E(I)*T1(1)/S2
112 T7(2)=-E(I)*T1(2)/S2
113 S7(1)=S7(1)+T7(1)
114 S7(2)=S7(2)+T7(2)
115 NEXT I
116 F(1)=S0*(Q2(1)*S7(1)-Q2(2)*S7(2))
117 F(2)=S0*(Q2(1)*S7(2)+Q2(2)*S7(1))
118 IF (M=0) THEN 122
119 IF (S0=1) THEN 121
120 F(1)=2+F(1)
121 EXIT SUB
122 F(1)=S0-F(1)
123 F(2)=-F(2)
124 EXIT SUB
125 T7(1)=2*R*R1(1)
126 T7(2)=2*R*R1(2)
127 S7(1)=T7(1)
128 S7(2)=T7(2)
129 M7=0
130 M7=M7+1
131 N7=2*M7+1
132 T1(1)=T7(1)*R2(1)-T7(2)*R2(2)
133 T1(2)=T7(1)*R2(2)+T7(2)*R2(1)
134 T7(1)=-T1(1)/M7
135 T7(2)=-T1(2)/M7
136 T1(1)=T7(1)/N7
137 T1(2)=T7(2)/N7
138 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 140
139 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 143
140 S7(1)=S7(1)+T1(1)
141 S7(2)=S7(2)+T1(2)
142 GOTO 130
143 F(1)=S0*S7(1)
144 F(2)=S0*S7(2)
145 IF (M<>0) THEN 147
146 EXIT SUB
147 F(1)=S0-F(1)
148 F(2)=-F(2)
149 EXIT SUB
150 END SUB
```

CERF

```
151 OPTION NOLET
152 DIM Z(2),F(2)
153 Z(1)=1
154 Z(2)=0
155 M=1
156 CALL CERF (M,Z(),F())
157 Q=1-F(1)
158 PRINT USING "+#.##########^^^^": Q
159 PRINT USING "+#.##########^^^^": F(1)
160 PRINT USING "+#.##########^^^^": F(2)
161 M=0
162 CALL CERF (M,Z(),F())
163 PRINT USING "+#.##########^^^^": 1-F(1)
164 PRINT USING "+#.##########^^^^": -F(2)
165 END
```

```
1 SUB PSWKVP (K,H,X,Y,Z,P)
2 !************************************
3 !POINT SOURCE WAKE VELOCITY POTENTIAL
4 !************************************
5 !K = KAPPA ZERO
6 !H = DEPTH OF SOURCE
7 !X = DISTANCE FORWARD
8 !Y = DISTANCE TO RIGHT
9 !Z = DISTANCE DOWNWARD
10 !P = VELOCITY POTENTIAL
11 OPTION NOLET
12 IF (H1=1) THEN 16
13 H1=1
14 DIM D1(2),E1(2)
15 DIM S3(2),T3(2)
16 X1=K*X
17 Y1=ABS(K*Y)
18 M1=K*(Z-H)
19 S2=X1*X1+Y1*Y1+M1*M1
20 S1=SQR(S2)
21 IF (S1>0) THEN 24
22 P=0
23 GOTO 25
24 P=K/S1
25 M1=K*(Z+H)
26 S2=X1*X1+Y1*Y1+M1*M1
27 S1=SQR(S2)
28 P=P-K/S1
29 S2=X1*X1+Y1*Y1
30 S1=SQR(S2)
31 Q1=SQR(M1)
32 N=1+27/Q1
33 IF (S1=0) THEN 36
34 Q3=Y1/(S1*Q1)
35 N=N+(S1/M1)*(((7*Q3)+54)*Q3+7)
36 N=N/2
37 N=2*IP(N)
38 S3(1)=0
39 S3(2)=0
40 M3=0
41 FOR L=1 TO N
42 Q1=((M3+.5)*.5*PI)/N
43 C1=+COS(Q1)
44 S1=+SIN(Q1)
45 C2=C1*C1
46 D1(1)=+M1/C2
47 D1(2)=-(X1*C1+Y1*S1)/C2
48 M0=1
49 CALL EXPLI (M0,D1(),E1())
50 IF (D1(2)<=0) THEN 56
```

```
51 Q3=EXP(-D1(1))
52 T3(1)=2*PI*Q3*COS(-D1(2))
53 T3(2)=2*PI*Q3*SIN(-D1(2))
54 E1(1)=E1(1)-T3(2)
55 E1(2)=E1(2)+T3(1)
56 T3(1)=E1(1)/C2
57 T3(2)=E1(2)/C2
58 S3(1)=S3(1)+T3(1)
59 S3(2)=S3(2)+T3(2)
60 S1=-S1
61 D1(1)=+M1/C2
62 D1(2)=-(X1*C1+Y1*S1)/C2
63 M0=1
64 CALL EXPLI (M0,D1(),E1())
65 IF (D1(2)<=0) THEN 71
66 Q3=EXP(-D1(1))
67 T3(1)=2*PI*Q3*COS(-D1(2))
68 T3(2)=2*PI*Q3*SIN(-D1(2))
69 E1(1)=E1(1)-T3(2)
70 E1(2)=E1(2)+T3(1)
71 T3(1)=E1(1)/C2
72 T3(2)=E1(2)/C2
73 S3(1)=S3(1)+T3(1)
74 S3(2)=S3(2)+T3(2)
75 M3=M3+1
76 NEXT L
77 Q=K*S3(1)/N
78 P=P+Q
79 EXIT SUB
80 END SUB
81 SUB EXPLI (M,A(),F())
82 !************************************
83 !SUBROUTINE FOR EXPONENTIAL INTEGRAL
84 !************************************
85 ! A = ARGUMENT (COMPLEX ARRAY)
86 ! F = INTEGRAL (COMPLEX ARRAY)
87 OPTION NOLET
88 IF (H2=1) THEN 137
89 H2=1
90 DIM D(18),E(18),L1(2),T1(2)
91 DIM Q2(2),R1(2),S7(2),T7(2)
92 DATA 0.5772156649015329E00
93 READ G
94  DATA 0.000000000000000E00
95  DATA 3.111059570865283E-2
96  DATA 1.036612605391116E-1
97  DATA 2.165323352445536E-1
98  DATA 3.699314279601916E-1
99  DATA 5.667662599905892E-1
100 DATA 8.140420663247483E-1
```

```
101 DATA 1.123842475408128E00
102 DATA 1.514004781485123E00
103 DATA 2.008867950322836E00
104 DATA 2.640524118235915E00
105 DATA 3.450984499333923E00
106 DATA 4.495833607632020E00
107 DATA 5.850582634098224E00
108 DATA 7.622735014633804E00
109 DATA 9.978145015845783E00
110 DATA 1.321220648964083E+1
111 DATA 1.803229483760214E+1
112 FOR I=1 TO 18
113 READ D(I)
114 NEXT I
115 DATA 8.501565161210931E-3
116 DATA 5.050374658490585E-2
117 DATA 8.368173689564071E-2
118 DATA 1.070475824176067E-1
119 DATA 1.204247190294619E-1
120 DATA 1.250966315822293E-1
121 DATA 1.223144352246853E-1
122 DATA 1.126214175539071E-1
123 DATA 9.634194073925819E-2
124 DATA 7.473984227575106E-2
125 DATA 5.085961359534413E-2
126 DATA 2.908227067736279E-2
127 DATA 1.322016405301009E-2
128 DATA 4.438029398290675E-3
129 DATA 9.926124789875764E-4
130 DATA 1.265797951120111E-4
131 DATA 7.021509082533505E-6
132 DATA 9.102815325646319E-8
133 FOR I=1 TO 18
134 READ E(I)
135 NEXT I
136 RESTORE
137 A2=A(1)*A(1)+A(2)*A(2)
138 IF (A2<=1) THEN 172
139 IF (A2>=1600) THEN 141
140 IF (-A(1)+.064*A(2)*A(2)>0) THEN 160 ELSE 172
141 S7(1)=0
142 S7(2)=0
143 Q2(1)=+A(1)/A2
144 Q2(2)=-A(2)/A2
145 T7(1)=Q2(1)
146 T7(2)=Q2(2)
147 M7=0
148 GOTO 156
149 M7=M7+1
150 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
```

```
151 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
152 T7(1)=M7*T1(1)
153 T7(2)=M7*T1(2)
154 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 156
155 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 199
156 S7(1)=S7(1)+T7(1)
157 S7(2)=S7(2)+T7(2)
158 IF (M7<40) THEN 149
159 GOTO 199
160 S7(1)=0
161 S7(2)=0
162 FOR I=1 TO 18
163 T1(1)=A(1)-D(I)
164 T1(2)=A(2)
165 S2=T1(1)*T1(1)+T1(2)*T1(2)
166 T7(1)=+E(I)*T1(1)/S2
167 T7(2)=-E(I)*T1(2)/S2
168 S7(1)=S7(1)+T7(1)
169 S7(2)=S7(2)+T7(2)
170 NEXT I
171 GOTO 199
172 L1(1)=.5*LOG(A2)
173 X7=-A(1)
174 Y7=-A(2)
175 GOSUB 215
176 L1(2)=Q7
177 R1(1)=G+L1(1)+A(1)
178 R1(2)=L1(2)+A(2)
179 S7(1)=0
180 S7(2)=0
181 T1(1)=A(1)
182 T1(2)=A(2)
183 M7=1
184 M7=M7+1
185 T7(1)=T1(1)*A(1)-T1(2)*A(2)
186 T7(2)=T1(1)*A(2)+T1(2)*A(1)
187 T1(1)=T7(1)/M7
188 T1(2)=T7(2)/M7
189 T7(1)=T1(1)/M7
190 T7(2)=T1(2)/M7
191 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 193
192 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 196
193 S7(1)=S7(1)+T7(1)
194 S7(2)=S7(2)+T7(2)
195 GOTO 184
196 S7(1)=S7(1)+R1(1)
197 S7(2)=S7(2)+R1(2)
198 GOTO 204
199 IF (M<>0) THEN 212
200 Q1=EXP(A(1))
```

```
201 Q2(1)=Q1*COS(A(2))
202 Q2(2)=Q1*SIN(A(2))
203 GOTO 208
204 IF (M=0) THEN 212
205 Q1=EXP(-A(1))
206 Q2(1)=Q1*COS(-A(2))
207 Q2(2)=Q1*SIN(-A(2))
208 T1(1)=Q2(1)*S7(1)-Q2(2)*S7(2)
209 T1(2)=Q2(1)*S7(2)+Q2(2)*S7(1)
210 S7(1)=T1(1)
211 S7(2)=T1(2)
212 F(1)=S7(1)
213 F(2)=S7(2)
214 EXIT SUB
215 Q7=ANGLE(X7,Y7)
216 RETURN
217 END SUB
218 OPTION NOLET
219 DIM A(2),F(2)
220 H1=0
221 H2=0
222 K=1
223 H=.1
224 X=-2
225 Y=0
226 Z=0
227 CALL PSWKVP (K,H,X,Y,Z,P)
228 PRINT USING "+#.###############^^^^": P
229 END
```

```
 1 SUB PSWKVF (K,H,X,Y,Z,U,V,W)
 2 !********************************
 3 !POINT SOURCE WAKE VELOCITY FIELD
 4 !********************************
 5 !K = KAPPA ZERO
 6 !H = DEPTH OF SOURCE
 7 !X = DISTANCE FORWARD
 8 !Y = DISTANCE TO RIGHT
 9 !Z = DISTANCE DOWNWARD
10 !U = COMPONENT FORWARD
11 !V = COMPONENT TO RIGHT
12 !W = COMPONENT DOWNWARD
13 OPTION NOLET
14 IF (H1=1) THEN 18
15 H1=1
16 DIM D1(2),E1(2)
17 DIM T3(2),U3(2),V3(2),W3(2)
18 X1=K*X
19 Y1=ABS(K*Y)
20 M1=K*(Z-H)
21 S2=X1*X1+Y1*Y1+M1*M1
22 S1=SQR(S2)
23 IF (S1>0) THEN 28
24 U=0
25 V=0
26 W=0
27 GOTO 31
28 U=K*K*X1/(S1*S2)
29 V=K*K*Y1/(S1*S2)
30 W=K*K*M1/(S1*S2)
31 M1=K*(Z+H)
32 S2=X1*X1+Y1*Y1+M1*M1
33 S1=SQR(S2)
34 U=U-K*K*X1/(S1*S2)
35 V=V-K*K*Y1/(S1*S2)
36 W=W-K*K*M1/(S1*S2)
37 S2=X1*X1+Y1*Y1
38 S1=SQR(S2)
39 Q1=SQR(M1)
40 N=1+27/Q1
41 IF (S1=0) THEN 44
42 Q3=Y1/(S1*Q1)
43 N=N+(S1/M1)*(((7*Q3)+54)*Q3+7)
44 N=N/2
45 N=2*IP(N)
46 U3(1)=0
47 U3(2)=0
48 V3(1)=0
49 V3(2)=0
50 W3(1)=0
```

```
51 W3(2)=0
52 M3=0
53 FOR L=1 TO N
54 Q1=((M3+.5)*.5*PI)/N
55 C1=+COS(Q1)
56 S1=+SIN(Q1)
57 C2=C1*C1
58 D1(1)=+M1/C2
59 D1(2)=-(X1*C1+Y1*S1)/C2
60 M0=1
61 CALL XPNLI (M0,D1(),E1())
62 IF (D1(2)<=0) THEN 68
63 Q3=EXP(-D1(1))
64 T3(1)=2*PI*Q3*COS(-D1(2))
65 T3(2)=2*PI*Q3*SIN(-D1(2))
66 E1(1)=E1(1)-T3(2)
67 E1(2)=E1(2)+T3(1)
68 T3(1)=E1(1)/(C2*C2)
69 T3(2)=E1(2)/(C2*C2)
70 U3(1)=U3(1)+C1*T3(1)
71 U3(2)=U3(2)+C1*T3(2)
72 V3(1)=V3(1)+S1*T3(1)
73 V3(2)=V3(2)+S1*T3(2)
74 W3(1)=W3(1)+T3(1)
75 W3(2)=W3(2)+T3(2)
76 S1=-S1
77 D1(1)=+M1/C2
78 D1(2)=-(X1*C1+Y1*S1)/C2
79 M0=1
80 CALL XPNLI (M0,D1(),E1())
81 IF (D1(2)<=0) THEN 87
82 Q3=EXP(-D1(1))
83 T3(1)=2*PI*Q3*COS(-D1(2))
84 T3(2)=2*PI*Q3*SIN(-D1(2))
85 E1(1)=E1(1)-T3(2)
86 E1(2)=E1(2)+T3(1)
87 T3(1)=E1(1)/(C2*C2)
88 T3(2)=E1(2)/(C2*C2)
89 U3(1)=U3(1)+C1*T3(1)
90 U3(2)=U3(2)+C1*T3(2)
91 V3(1)=V3(1)+S1*T3(1)
92 V3(2)=V3(2)+S1*T3(2)
93 W3(1)=W3(1)+T3(1)
94 W3(2)=W3(2)+T3(2)
95 M3=M3+1
96 NEXT L
97 U=U+K*K*U3(2)/N
98 V=V+K*K*V3(2)/N
99 W=W+K*K*W3(1)/N
100 EXIT SUB
```

PSWKVF

```
101 END SUB
102 SUB XPNLI (M,A(),F())
103 !****************************
104 !SPECIAL EXPONENTIAL INTEGRAL
105 !****************************
106 !A = ARGUMENT (COMPLEX ARRAY)
107 !F = INTEGRAL (COMPLEX ARRAY)
108 OPTION NOLET
109 IF (H2=1) THEN 158
110 H2=1
111 DIM D(18),E(18),L1(2),T1(2)
112 DIM Q2(2),R1(2),S7(2),T7(2)
113 DATA 0.5772156649015329E00
114 READ G
115 DATA 0.000000000000000E00
116 DATA 3.111059570865283E-2
117 DATA 1.036612605391116E-1
118 DATA 2.165323352445536E-1
119 DATA 3.699314279601916E-1
120 DATA 5.667662599905892E-1
121 DATA 8.140420663247483E-1
122 DATA 1.123842475408128E00
123 DATA 1.514004781485123E00
124 DATA 2.008867950322836E00
125 DATA 2.640524118235915E00
126 DATA 3.450984499333923E00
127 DATA 4.495833607632020E00
128 DATA 5.850582634098224E00
129 DATA 7.622735014633804E00
130 DATA 9.978145015845783E00
131 DATA 1.321220648964083E+1
132 DATA 1.803229483760214E+1
133 FOR I=1 TO 18
134 READ D(I)
135 NEXT I
136 DATA-9.914984348387891E-1
137 DATA 5.050374658490585E-2
138 DATA 8.368173689564071E-2
139 DATA 1.070475824176067E-1
140 DATA 1.204247190294619E-1
141 DATA 1.250966315822293E-1
142 DATA 1.223144352246853E-1
143 DATA 1.126214175539071E-1
144 DATA 9.634194073925819E-2
145 DATA 7.473984227575106E-2
146 DATA 5.085961359534413E-2
147 DATA 2.908227067736279E-2
148 DATA 1.322016405301009E-2
149 DATA 4.438029398290675E-3
150 DATA 9.926124789875764E-4
```

```
151 DATA 1.265797951120111E-4
152 DATA 7.021509082533505E-6
153 DATA 9.102815325646319E-8
154 FOR I=1 TO 18
155 READ E(I)
156 NEXT I
157 RESTORE
158 A2=A(1)*A(1)+A(2)*A(2)
159 IF (A2<=1) THEN 193
160 IF (A2>=1600) THEN 163
161 IF (-A(1)+.064*A(2)*A(2)>0) THEN 181
162 GOTO 193
163 S7(1)=0
164 S7(2)=0
165 Q2(1)=+A(1)/A2
166 Q2(2)=-A(2)/A2
167 T7(1)=Q2(1)
168 T7(2)=Q2(2)
169 M7=0
170 M7=M7+1
171 T1(1)=T7(1)*Q2(1)-T7(2)*Q2(2)
172 T1(2)=T7(1)*Q2(2)+T7(2)*Q2(1)
173 T7(1)=M7*T1(1)
174 T7(2)=M7*T1(2)
175 IF (ABS(S7(1))+ABS(T7(1))>ABS(S7(1))) THEN 177
176 IF (ABS(S7(2))+ABS(T7(2))=ABS(S7(2))) THEN 220
177 S7(1)=S7(1)+T7(1)
178 S7(2)=S7(2)+T7(2)
179 IF (M7<40) THEN 170
180 GOTO 220
181 S7(1)=0
182 S7(2)=0
183 FOR I=1 TO 18
184 T1(1)=A(1)-D(I)
185 T1(2)=A(2)
186 S2=T1(1)*T1(1)+T1(2)*T1(2)
187 T7(1)=+E(I)*T1(1)/S2
188 T7(2)=-E(I)*T1(2)/S2
189 S7(1)=S7(1)+T7(1)
190 S7(2)=S7(2)+T7(2)
191 NEXT I
192 GOTO 220
193 L1(1)=.5*LOG(A2)
194 X7=-A(1)
195 Y7=-A(2)
196 GOSUB 247
197 L1(2)=Q7
198 R1(1)=G+L1(1)+A(1)
199 R1(2)=L1(2)+A(2)
200 S7(1)=0
```

```
201 S7(2)=0
202 T7(1)=A(1)
203 T7(2)=A(2)
204 M7=1
205 M7=M7+1
206 T1(1)=T7(1)*A(1)-T7(2)*A(2)
207 T1(2)=T7(1)*A(2)+T7(2)*A(1)
208 T7(1)=T1(1)/M7
209 T7(2)=T1(2)/M7
210 T1(1)=T7(1)/M7
211 T1(2)=T7(2)/M7
212 IF (ABS(S7(1))+ABS(T1(1))>ABS(S7(1))) THEN 214
213 IF (ABS(S7(2))+ABS(T1(2))=ABS(S7(2))) THEN 217
214 S7(1)=S7(1)+T1(1)
215 S7(2)=S7(2)+T1(2)
216 GOTO 205
217 S7(1)=S7(1)+R1(1)
218 S7(2)=S7(2)+R1(2)
219 GOTO 229
220 IF (M<>0) THEN 244
221 Q1=EXP(A(1))
222 Q2(1)=Q1*COS(A(2))
223 Q2(2)=Q1*SIN(A(2))
224 T1(1)=Q2(1)*S7(1)-Q2(2)*S7(2)
225 T1(2)=Q2(1)*S7(2)+Q2(2)*S7(1)
226 S7(1)=T1(1)
227 S7(2)=T1(2)
228 GOTO 244
229 Q1=EXP(-A(1))
230 T7(1)=COS(-A(2))
231 T7(2)=SIN(-A(2))
232 Q2(1)=+A(1)/A2
233 Q2(2)=-A(2)/A2
234 IF (M<>0) THEN 240
235 T1(1)=(T7(1)*Q2(1)+T7(2)*Q2(2))/Q1
236 T1(2)=(T7(1)*Q2(2)-T7(2)*Q2(1))/Q1
237 S7(1)=S7(1)-T1(1)
238 S7(2)=S7(2)-T1(2)
239 GOTO 244
240 T1(1)=(T7(1)*S7(1)-T7(2)*S7(2))*Q1
241 T1(2)=(T7(1)*S7(2)+T7(2)*S7(1))*Q1
242 S7(1)=T1(1)-Q2(1)
243 S7(2)=T1(2)-Q2(2)
244 F(1)=S7(1)
245 F(2)=S7(2)
246 EXIT SUB
247 Q7=ANGLE(X7,Y7)
248 RETURN
249 END SUB
250 OPTION NOLET
```

```
251 DIM A(2),F(2)
252 H1=0
253 H2=0
254 K=1
255 H=.1
256 X=-2
257 Y=0
258 Z=0
259 CALL PSWKVF (K,H,X,Y,Z,U,V,W)
260 PRINT USING " +#.###############^^^^": U,V,W
261 END
```

```
1 SUB LGRNX (N,A(),W())
2 !***********************************
3 !BASIC LAGRANGE POLYNOMIAL EXPANSION
4 !***********************************
5 !N = NUMBER
6 !A = ARGUMENTS          N ARRAY
7 !W = COEFFICIENTS      (N+1)*N ARRAY
8 !POLYNOMIAL F IN (N+1)TH ROW
9  !COEFFICIENTS ALONG ROWS
10 !ARGUMENTS DOWN COLUMNS
11 !
12 OPTION NOLET
13 FOR I=1 TO N
14 K=(I-1)*N
15 L=K+I
16 W(L)=1
17 S=0
18 FOR J=1 TO I
19 K=K+1
20 L=K+N
21 W(L)=S-A(I)*W(K)
22 S=W(K)
23 NEXT J
24 NEXT I
25 FOR I=1 TO N
26 K=N+N*N
27 L=I*N
28 W(L)=1
29 FOR J=2 TO N
30 S=A(I)*W(L)
31 L=L-1
32 W(L)=S+W(K)
33 K=K-1
34 NEXT J
35 NEXT I
36 FOR I=1 TO N
37 T=1
38 FOR J=1 TO N
39 IF (J=I) THEN 42
40 D=A(I)-A(J)
41 T=D*T
42 NEXT J
43 K=(I-1)*N
44 FOR J=1 TO N
45 K=K+1
46 W(K)=W(K)/T
47 NEXT J
48 NEXT I
49 EXIT SUB
50 END SUB
```

LGRNX

```
51 OPTION NOLET
52 DIM A(5),W(30),C(5)
53 A(1)=-1
54 A(2)=-.5
55 A(3)=0
56 A(4)=+.5
57 A(5)=+1
58 N=5
59 CALL LGRNX (N,A(),W())
60 FOR I=1 TO 5
61 PRINT USING "+#.###########^^^^": W(25+I)
62 NEXT I
63 C(1)=0
64 C(2)=.5625
65 C(3)=1
66 C(4)=.5625
67 C(5)=0
68 FOR I=1 TO 5
69 S=0
70 FOR J=1 TO 5
71 K=I+5*J-5
72 S=S+C(J)*W(K)
73 NEXT J
74 PRINT USING "+#.###########^^^^": S
75 NEXT I
76 END
```

LGRNX

```
1 SUB CFOURX (N,M,W())
2 !********************************
3 !FOURIER SERIES EXPANSION
4 !********************************
5 !N = NUMBER OF DATA
6 !M = NUMBER OF COEFFICIENTS
7 !W = MATRIX OF COEFFICIENTS (N*M ARRAY)
8 !
9  !COEFFICIENTS ALONG ROW
10 !ARGUMENTS DOWN COLUMN
11 !
12 OPTION NOLET
13 P=4*ATN(1)
14 N7=N
15 I7=0
16 L=0
17 FOR I=1 TO N
18 Q=2*P*(I7/N7)
19 W(L+1)=+COS(Q)
20 W(L+2)=+SIN(Q)
21 I7=I7+1
22 L=L+2*M
23 NEXT I
24 IF (M=1) THEN 51
25 I7=.5
26 L=2
27 FOR I=1 TO N
28 Q=2*P*(I7/N7)
29 W(L+1)=+COS(Q)
30 W(L+2)=+SIN(Q)
31 I7=I7+1
32 L=L+2*M
33 NEXT I
34 IF (M=2) THEN 51
35 Q=M-2
36 J7=2
37 FOR J=3 TO M
38 I7=.5
39 L=2*(J-1)
40 FOR I=1 TO N
41 Q7=I7*J7
42 Q7=Q7-N7*SGN(Q7)*INT(ABS(Q7/N7))
43 K=INT(4*Q7+2*Q*SGN(Q7)*INT(ABS(Q7)))
44 W(L+1)=W(K+1)
45 W(L+2)=W(K+2)
46 I7=I7+1
47 L=L+2*M
48 NEXT I
49 J7=J7+1
50 NEXT J
```

CFOURX

```
51 L=0
52 FOR I=1 TO N
53 W(L+1)=1
54 W(L+2)=0
55 L=L+2*M
56 NEXT I
57 EXIT SUB
58 END SUB
59 OPTION NOLET
60 DIM W(80)
61 N=8
62 M=5
63 CALL CFOURX (N,M,W())
64 C=0
65 S=0
66 FOR K=1 TO N
67 L=1+2*M*(K-1)
68 C=C+W(L)*W(L)
69 S=S+W(L+1)*W(L+1)
70 GOTO 71
71 PRINT USING " +#.###############^^^^": W(L),W(L+1)
72 NEXT K
73 PRINT USING " +#.###############^^^^": C,S
74 END
```

```
1 SUB CFOURV (D,Q,M,C(),F())
2 !*********************************
3 !FOURIER SERIES EVALUATION
4 !*********************************
5 !D = MODE OF DIFFERENTIATION
6 !Q = ANGULAR ARGUMENT
7 !M = NUMBER OF COEFFICIENTS
8 !C = ARRAY OF COEFFICIENTS (2*M ARRAY)
9 !F = COMPLEX FUNCTION      (2 ARRAY)
10 !D = 0 FOR FUNCTION
11 !D = 1 FOR FIRST DERIVATIVE
12 !D = 2 FOR SECOND DERIVATIVE
13 OPTION NOLET
14 IF (H=1) THEN 17
15 H=1
16 DIM Z(2),T7(2)
17 Z(1)=COS(Q)
18 Z(2)=SIN(Q)
19 F(1)=0
20 F(2)=0
21 L=2*M
22 IF (D=0) THEN 25
23 IF (D=1) THEN 33
24 IF (D=2) THEN 44
25 FOR K=1 TO M
26 T7(1)=Z(1)*F(1)-Z(2)*F(2)
27 T7(2)=Z(1)*F(2)+Z(2)*F(1)
28 F(1)=T7(1)+C(L-1)
29 F(2)=T7(2)+C(L)
30 L=L-2
31 NEXT K
32 EXIT SUB
33 IF (M<=1) THEN 43
34 Q7=M
35 FOR K=1 TO M
36 Q7=Q7-1
37 T7(1)=Z(1)*F(1)-Z(2)*F(2)
38 T7(2)=Z(1)*F(2)+Z(2)*F(1)
39 F(1)=T7(1)-Q7*C(L)
40 F(2)=T7(2)+Q7*C(L-1)
41 L=L-2
42 NEXT K
43 EXIT SUB
44 IF (M<=1) THEN 54
45 Q7=M
46 FOR K=1 TO M
47 Q7=Q7-1
48 T7(1)=Z(1)*F(1)-Z(2)*F(2)
49 T7(2)=Z(1)*F(2)+Z(2)*F(1)
50 F(1)=T7(1)-Q7*Q7*C(L-1)
```

```
51 F(2)=T7(2)-Q7*Q7*C(L)
52 L=L-2
53 NEXT K
54 EXIT SUB
55 END SUB
56 OPTION NOLET
57 DIM C(10),F(2)
58 D=0
59 M=5
60 C(1)=1
61 C(2)=0
62 C(3)=0
63 C(4)=0
64 C(5)=0
65 C(6)=0
66 C(7)=0
67 C(8)=0
68 C(9)=0
69 C(10)=0
70 P=.5*ATN(1)
71 FOR N=0 TO 16
72 Q=N*P
73 CALL CFOURV (D,Q,M,C(),F())
74 PRINT USING " +#.###############^^^^": F(1),F(2)
75 NEXT N
76 END
```

```
1 SUB CXFFT (M,L,A())
2 !***********************************
3 !COMPLEX FAST FOURIER TRANSFORM
4 !***********************************
5 !M = MODE OF OPERATION
6 !L = LOG2 N
7 !A = DATA OR COEFFICIENTS (2N ARRAY)
8 !
9  !M = -1 FOR DATA TO COEFFICIENTS
10 !M = +1 FOR COEFFICIENTS TO DATA
11 !
12 OPTION NOLET
13 IF (H=1) THEN 16
14 H=1
15 DIM T7(2)
16 IF (L<=0) THEN 80
17 N=1
18 FOR N7=1 TO L
19 N=N+N
20 NEXT N7
21 A0=N
22 N2=N/2
23 FOR L7=1 TO L
24 K7=0
25 B0=K7/N2
26 R0=0
27 FOR J7=1 TO L
28 I0=IP(B0/2)
29 R0=2*R0+(B0-2*I0)
30 B0=I0
31 NEXT J7
32 Q0=M*R0
33 Q0=2*PI*(Q0/A0)
34 C0=COS(Q0)
35 S0=SIN(Q0)
36 M7=2*K7
37 N7=M7+2*N2
38 FOR I7=1 TO N2
39 T7(1)=C0*A(N7+1)-S0*A(N7+2)
40 T7(2)=C0*A(N7+2)+S0*A(N7+1)
41 A(N7+1)=A(M7+1)-T7(1)
42 A(N7+2)=A(M7+2)-T7(2)
43 A(M7+1)=A(M7+1)+T7(1)
44 A(M7+2)=A(M7+2)+T7(2)
45 K7=K7+1
46 M7=M7+2
47 N7=N7+2
48 NEXT I7
49 K7=K7+N2
50 IF (K7<N) THEN 25
```

```
51 N2=N2/2
52 NEXT L7
53 K7=0
54 FOR I7=1 TO N
55 B0=K7
56 R0=0
57 FOR J7=1 TO L
58 I0=IP(B0/2)
59 R0=2*R0+(B0-2*I0)
60 B0=I0
61 NEXT J7
62 M7=2*K7
63 N7=2*R0
64 IF (N7<=M7) THEN 71
65 T7(1)=A(M7+1)
66 T7(2)=A(M7+2)
67 A(M7+1)=A(N7+1)
68 A(M7+2)=A(N7+2)
69 A(N7+1)=T7(1)
70 A(N7+2)=T7(2)
71 K7=K7+1
72 NEXT I7
73 IF (M=1) THEN 80
74 M7=0
75 FOR I7=1 TO N
76 A(M7+1)=A(M7+1)/A0
77 A(M7+2)=A(M7+2)/A0
78 M7=M7+2
79 NEXT I7
80 EXIT SUB
81 END SUB
82 OPTION NOLET
83 DIM A(16),C(16)
84 N=8
85 X=1
86  A(1)=COS(X*0*PI/N)
87  A(2)=SIN(X*0*PI/N)
88  A(3)=COS(X*2*PI/N)
89  A(4)=SIN(X*2*PI/N)
90  A(5)=COS(X*4*PI/N)
91  A(6)=SIN(X*4*PI/N)
92  A(7)=COS(X*6*PI/N)
93  A(8)=SIN(X*6*PI/N)
94  A(9)=COS(X*8*PI/N)
95 A(10)=SIN(X*8*PI/N)
96 A(11)=COS(X*10*PI/N)
97 A(12)=SIN(X*10*PI/N)
98 A(13)=COS(X*12*PI/N)
99 A(14)=SIN(X*12*PI/N)
100 A(15)=COS(X*14*PI/N)
```

```
101 A(16)=SIN(X*14*PI/N)
102 C(1)=0
103 C(2)=0
104 C(3)=1
105 C(4)=0
106 C(5)=0
107 C(6)=0
108 C(7)=0
109 C(8)=0
110 C(9)=0
111 C(10)=0
112 C(11)=0
113 C(12)=0
114 C(13)=0
115 C(14)=0
116 C(15)=0
117 C(16)=0
118 L=3
119 FOR K=1 TO 8
120 N=2*K-1
121 A(N)=C(N)
122 A(N+1)=C(N+1)
123 GOTO 125
124 PRINT USING " #.###############^^^^": C(N),C(N+1)
125 NEXT K
126 M=+1
127 CALL CXFFT (M,L,A())
128 GOTO 129
129 FOR K=1 TO 8
130 N=2*K-1
131 PRINT USING " #.###############^^^^": A(N),A(N+1)
132 NEXT K
133 M=-1
134 CALL CXFFT (M,L,A())
135 GOTO 136
136 FOR K=1 TO 8
137 N=2*K-1
138 PRINT USING " #.###############^^^^": A(N),A(N+1)
139 NEXT K
140 END
```

```
1 SUB CXRT (D,E,Z(),N,R())
2 !********************************
3 !SUBROUTINE FOR COMPLEX ROOTS
4 !********************************
5 !D = DELTA, E = EPSILON
6 !Z = INITIAL ARGUMENT
7 !N = NUMBER OF ROOTS
8 !R = ARRAY OF ROOTS
9 !SUB FXN (M,Z(),F(),D(),S())
10 !Z = ARGUMENT
11 !F = FUNCTION          (M=0)
12 !D = FIRST DERIVATIVE  (M=1)
13 !S = SECOND DERIVATIVE (M=2)
14 OPTION NOLET
15 IF (H=1) THEN 20
16 H=1
17 DIM D1(2),E1(2),Z0(2),Z1(2),Z2(2),Z7(2)
18 DIM F0(2),F1(2),F2(2),W0(2),W1(2),W2(2)
19 DIM D7(2),P7(2),R7(2),S7(2),T7(2)
20 K=0
21 E0=E
22 Z7(1)=Z(1)
23 Z7(2)=Z(2)
24 Z0(1)=Z(1)
25 Z0(2)=Z(2)
26 L=0
27 D0=D
28 M0=2
29 CALL FXN (M0,Z0(),F0(),F1(),F2())
30 GOSUB 186
31 Q1=P7(1)*P7(1)+P7(2)*P7(2)
32 W0(1)=+(F0(1)*P7(1)+F0(2)*P7(2))/Q1
33 W0(2)=-(F0(1)*P7(2)-F0(2)*P7(1))/Q1
34 T7(1)=F1(1)-F0(1)*R7(1)+F0(2)*R7(2)
35 T7(2)=F1(2)-F0(1)*R7(2)-F0(2)*R7(1)
36 W1(1)=+(T7(1)*P7(1)+T7(2)*P7(2))/Q1
37 W1(2)=-(T7(1)*P7(2)-T7(2)*P7(1))/Q1
38 T7(1)=S7(1)+R7(1)*R7(1)-R7(2)*R7(2)
39 T7(2)=S7(2)+2*R7(1)*R7(2)
40 W2(1)=F0(1)*T7(1)-F0(2)*T7(2)
41 W2(2)=F0(1)*T7(2)+F0(2)*T7(1)
42 T7(1)=F2(1)-2*(F1(1)*R7(1)-F1(2)*R7(2))+W2(1)
43 T7(2)=F2(2)-2*(F1(1)*R7(2)+F1(2)*R7(1))+W2(2)
44 W2(1)=+(T7(1)*P7(1)+T7(2)*P7(2))/Q1
45 W2(2)=-(T7(1)*P7(2)-T7(2)*P7(1))/Q1
46 D2=W1(1)*W1(1)+W1(2)*W1(2)
47 IF (D2=0) THEN 168
48 Z2(1)=-(W0(1)*W1(1)+W0(2)*W1(2))/D2
49 Z2(2)=+(W0(1)*W1(2)-W0(2)*W1(1))/D2
50 Z1(1)=Z7(1)
```

```
51 Z1(2)=Z7(2)
52 Z7(1)=Z0(1)+Z2(1)
53 Z7(2)=Z0(2)+Z2(2)
54 E1(1)=Z7(1)-Z1(1)
55 E1(2)=Z7(2)-Z1(2)
56 IF (L=1) THEN 59
57 L=1
58 GOTO 62
59 Q1=E1(1)*E1(1)+E1(2)*E1(2)
60 Q2=D1(1)*D1(1)+D1(2)*D1(2)
61 IF (Q1<Q2) THEN 107
62 Z2(1)=W1(1)*W1(1)-W1(2)*W1(2)-2*(W0(1)*W2(1)-W0(2)*W2(2))
63 Z2(2)=2*W1(1)*W1(2)-2*(W0(1)*W2(2)+W0(2)*W2(1))
64 GOSUB 214
65 IF (W1(1)*Z1(1)+W1(2)*Z1(2)>0) THEN 68
66 Z1(1)=-Z1(1)
67 Z1(2)=-Z1(2)
68 T7(1)=W1(1)+Z1(1)
69 T7(2)=W1(2)+Z1(2)
70 Q1=T7(1)*T7(1)+T7(2)*T7(2)
71 IF (Q1=0) THEN 168
72 Z2(1)=-2*(W0(1)*T7(1)+W0(2)*T7(2))/Q1
73 Z2(2)=+2*(W0(1)*T7(2)-W0(2)*T7(1))/Q1
74 S2=Z2(1)*Z2(1)+Z2(2)*Z2(2)
75 R2=SQR(S2)
76 D0=D+D0*R2/(D0+R2)
77 IF (R2<>0) THEN 86
78 Z2(1)=Z0(1)-Z(1)
79 Z2(2)=Z0(2)-Z(2)
80 S2=Z2(1)*Z2(1)+Z2(2)*Z2(2)
81 R2=SQR(S2)
82 IF (R2<>0) THEN 86
83 D1(1)=1
84 D1(2)=0
85 GOTO 88
86 D1(1)=Z2(1)/R2
87 D1(2)=Z2(2)/R2
88 Z1(1)=Z0(1)
89 Z1(2)=Z0(2)
90 Z0(1)=Z0(1)+D0*D1(1)
91 Z0(2)=Z0(2)+D0*D1(2)
92 GOTO 95
93 Z0(1)=Z0(1)+E0*D1(1)
94 Z0(2)=Z0(2)+E0*D1(2)
95 I=0
96 GOTO 102
97 I=I+1
98 T7(1)=Z0(1)-R(2*I-1)
99 T7(2)=Z0(2)-R(2*I)
100 Q1=T7(1)*T7(1)+T7(2)*T7(2)
```

```
101 IF (Q1<=.5*E0*E0) THEN 93
102 IF (I<K) THEN 97
103 D1(1)=Z0(1)-Z1(1)
104 D1(2)=Z0(2)-Z1(2)
105 D0=D0+D0
106 GOTO 28
107 Z0(1)=Z7(1)
108 Z0(2)=Z7(2)
109 L=0
110 M0=1
111 CALL FXN (M0,Z0(),F0(),F1(),F2())
112 GOSUB 186
113 Q1=P7(1)*P7(1)+P7(2)*P7(2)
114 W0(1)=+(F0(1)*P7(1)+F0(2)*P7(2))/Q1
115 W0(2)=-(F0(1)*P7(2)-F0(2)*P7(1))/Q1
116 T7(1)=F1(1)-F0(1)*R7(1)+F0(2)*R7(2)
117 T7(2)=F1(2)-F0(1)*R7(2)-F0(2)*R7(1)
118 W1(1)=+(T7(1)*P7(1)+T7(2)*P7(2))/Q1
119 W1(2)=-(T7(1)*P7(2)-T7(2)*P7(1))/Q1
120 IF (W0(1)<>0) THEN 122
121 IF (W0(2)<>0) THEN 160
122 S1=W1(1)*W1(1)+W1(2)*W1(2)
123 IF (S1=0) THEN 232
124 Z2(1)=-(W0(1)*W1(1)+W0(2)*W1(2))/S1
125 Z2(2)=+(W0(1)*W1(2)-W0(2)*W1(1))/S1
126 P7(1)=Z0(1)+Z2(1)
127 P7(2)=Z0(2)+Z2(2)
128 E1(1)=P7(1)-Z0(1)
129 E1(2)=P7(2)-Z0(2)
130 Q1=ABS(E1(1))+ABS(E1(2))
131 Q2=ABS(D1(1))+ABS(D1(2))
132 S1=1
133 IF (Q2<=1.1*Q1) THEN 135
134 S1=Q2/(Q2-Q1)
135 Z2(1)=Z0(1)+S1*E1(1)
136 Z2(2)=Z0(2)+S1*E1(2)
137 IF (L=1) THEN 140
138 L=1
139 GOTO 153
140 Q1=ABS(Z2(1)-Z1(1))
141 Q2=ABS(Z2(2)-Z1(2))
142 D2=10*D
143 IF (D2+Q1>D2) THEN 148
144 IF (D2+Q2>D2) THEN 148
145 Z0(1)=Z2(1)
146 Z0(2)=Z2(2)
147 GOTO 160
148 Q1=E1(1)*E1(1)+E1(2)*E1(2)
149 Q2=D1(1)*D1(1)+D1(2)*D1(2)
150 IF (Q1<Q2) THEN 153
```

```
151 IF (Q2>=E0*E0) THEN 232
152 GOTO 160
153 D1(1)=E1(1)
154 D1(2)=E1(2)
155 Z0(1)=P7(1)
156 Z0(2)=P7(2)
157 Z1(1)=Z2(1)
158 Z1(2)=Z2(2)
159 GOTO 110
160 D2=10*D
161 IF (D2+ABS(Z0(1))=D2) THEN Z0(1)=0
162 IF (D2+ABS(Z0(2))=D2) THEN Z0(2)=0
163 K=K+1
164 R(2*K-1)=Z0(1)
165 R(2*K)=Z0(2)
166 IF (K<N) THEN 168
167 EXIT SUB
168 D1(1)=Z0(1)-Z(1)
169 D1(2)=Z0(2)-Z(2)
170 S2=D1(1)*D1(1)+D1(2)*D1(2)
171 R2=SQR(S2)
172 IF (R2<>0) THEN 175
173 Z0(1)=Z0(1)+E0
174 GOTO 177
175 Z0(1)=Z0(1)+E0*D1(1)/R2
176 Z0(2)=Z0(2)+E0*D1(2)/R2
177 I=0
178 GOTO 184
179 I=I+1
180 E1(1)=Z0(1)-R(2*I-1)
181 E1(2)=Z0(2)-R(2*I)
182 S2=E1(1)*E1(1)+E1(2)*E1(2)
183 IF (S2<=.5*E0*E0) THEN 172
184 IF (I<K) THEN 179
185 GOTO 26
186 P7(1)=1
187 P7(2)=0
188 R7(1)=0
189 R7(2)=0
190 S7(1)=0
191 S7(2)=0
192 I=0
193 GOTO 212
194 I=I+1
195 D7(1)=Z0(1)-R(2*I-1)
196 D7(2)=Z0(2)-R(2*I)
197 IF (D7(1)<>0) THEN 199
198 IF (D7(2)=0) THEN 212
199 T7(1)=P7(1)*D7(1)-P7(2)*D7(2)
200 T7(2)=P7(1)*D7(2)+P7(2)*D7(1)
```

```
201 P7(1)=T7(1)
202 P7(2)=T7(2)
203 IF (M0<=0) THEN 212
204 Q1=D7(1)*D7(1)+D7(2)*D7(2)
205 T7(1)=+D7(1)/Q1
206 T7(2)=-D7(2)/Q1
207 R7(1)=R7(1)+T7(1)
208 R7(2)=R7(2)+T7(2)
209 IF (M0=1) THEN 212
210 S7(1)=S7(1)+T7(1)*T7(1)-T7(2)*T7(2)
211 S7(2)=S7(2)+2*T7(1)*T7(2)
212 IF (I<K) THEN 194
213 RETURN
214 S2=Z2(1)*Z2(1)+Z2(2)*Z2(2)
215 R2=SQR(S2)
216 Z1(1)=0
217 Z1(2)=0
218 IF (Z2(1)=0) THEN 225
219 IF (Z2(1)>0) THEN 226
220 Z1(2)=SQR(-Z2(1)+R2)
221 Z1(1)=Z2(2)/Z1(2)
222 IF (Z2(2)>=0) THEN 228
223 Q1=-SQR(.5)
224 GOTO 229
225 IF (Z2(2)=0) THEN 231
226 Z1(1)=SQR(+Z2(1)+R2)
227 Z1(2)=Z2(2)/Z1(1)
228 Q1=+SQR(.5)
229 Z1(1)=Q1*Z1(1)
230 Z1(2)=Q1*Z1(2)
231 RETURN
232 E0=E0+E0
233 GOTO 168
234 END SUB
235 SUB FXN (M,Z(),F(),D(),S())
236 !******************************
237 !BASIC FUNCTION AND DERIVATIVES
238 !******************************
239 !M = MODE OF OPERATION
240 !Z = ARGUMENT
241 !F = FUNCTION
242 !D = FIRST DERIVATIVE
243 !S = SECOND DERIVATIVE
244 OPTION NOLET
245 IF (H=1) THEN 248
246 H=1
247 DIM A1(2),A2(2),A3(2),A4(2)
248 A1(1)=Z(1)
249 A1(2)=Z(2)
250 A2(1)=A1(1)*A1(1)-A1(2)*A1(2)
```

```
251 A2(2)=2*A1(1)*A1(2)
252 A3(1)=A1(1)*A2(1)-A1(2)*A2(2)
253 A3(2)=A1(1)*A2(2)+A1(2)*A2(1)
254 A4(1)=A1(1)*A3(1)-A1(2)*A3(2)
255 A4(2)=A1(1)*A3(2)+A1(2)*A3(1)
256 F(1)=1-2*A2(1)+A4(1)
257 F(2)=0-2*A2(2)+A4(2)
258 IF (M=0) THEN 264
259 D(1)=-4*A1(1)+4*A3(1)
260 D(2)=-4*A1(2)+4*A3(2)
261 IF (M=1) THEN 264
262 S(1)=-4+12*A2(1)
263 S(2)=-0+12*A2(2)
264 EXIT SUB
265 END SUB
266 OPTION NOLET
267 DIM Z(2),R(8)
268 D=.001
269 E=.000001
270 Z(1)=0
271 Z(2)=0
272 N=4
273 CALL CXRT (D,E,Z(),N,R())
274 FOR I=1 TO 4
275 PRINT USING "+#.######^^^^ ": R(2*I-1),R(2*I)
276 NEXT I
277 END
```

```
1 SUB MTRX (M,A(),I(),B(),J(),C(),K())
2 !********************************
3 !SUBROUTINE FOR MATRIX ARITHMETIC
4 !********************************
5  !A = ADDRESS OF MATRIX A
6  !I = SPECIFICATION OF MATRIX A
7  !B = ADDRESS OF MATRIX B
8  !J = SPECIFICATION OF MATRIX B
9  !C = ADDRESS OF MATRIX C
10 !K = SPECIFICATION OF MATRIX C
11 !K(1) = INTERVAL BETWEEN COLUMNS
12 !K(2) = NUMBER OF ROWS
13 !K(3) = NUMBER OF COLUMNS
14 !K(4) = INTERVAL BETWEEN ROWS
15 !
16 !M=0 FOR B=transfer A
17 !M=1 FOR B=transpose A
18 !M=2 FOR C=A+B
19 !M=3 FOR C=A-B
20 !M=4 FOR C=A*B
21 !M=5 FOR A=inverse A, B(1)=|A|
22 !
23 OPTION NOLET
24 IF (H=1) THEN 27
25 H=1
26 DIM R(256),S(256)
27 IF (M<=0) THEN 33
28 IF (M=1) THEN 47
29 IF (M=2) THEN 61
30 IF (M=3) THEN 78
31 IF (M=4) THEN 95
32 IF (M>=5) THEN 114
33 I2=I(2)
34 J2=I(3)
35 L1=I(1)
36 M1=J(1)
37 FOR I7=1 TO I2
38 L7=1+(I7-1)*I(4)
39 M7=1+(I7-1)*J(4)
40 FOR J7=1 TO J2
41 B(M7)=A(L7)
42 L7=L7+L1
43 M7=M7+M1
44 NEXT J7
45 NEXT I7
46 EXIT SUB
47 I2=I(2)
48 J2=I(3)
49 L1=I(1)
50 M1=J(4)
```

```
51 FOR I7=1 TO I2
52 L7=1+(I7-1)*I(4)
53 M7=1+(I7-1)*J(1)
54 FOR J7=1 TO J2
55 B(M7)=A(L7)
56 L7=L7+L1
57 M7=M7+M1
58 NEXT J7
59 NEXT I7
60 EXIT SUB
61 I2=I(2)
62 J2=I(3)
63 L1=I(1)
64 M1=J(1)
65 N1=K(1)
66 FOR I7=1 TO I2
67 L7=1+(I7-1)*I(4)
68 M7=1+(I7-1)*J(4)
69 N7=1+(I7-1)*K(4)
70 FOR J7=1 TO J2
71 C(N7)=A(L7)+B(M7)
72 L7=L7+L1
73 M7=M7+M1
74 N7=N7+N1
75 NEXT J7
76 NEXT I7
77 EXIT SUB
78 I2=I(2)
79 J2=I(3)
80 L1=I(1)
81 M1=J(1)
82 N1=K(1)
83 FOR I7=1 TO I2
84 L7=1+(I7-1)*I(4)
85 M7=1+(I7-1)*J(4)
86 N7=1+(I7-1)*K(4)
87 FOR J7=1 TO J2
88 C(N7)=A(L7)-B(M7)
89 L7=L7+L1
90 M7=M7+M1
91 N7=N7+N1
92 NEXT J7
93 NEXT I7
94 EXIT SUB
95 I2=I(2)
96 J2=J(3)
97 K2=I(3)
98 L2=I(1)
99 M2=J(4)
100 FOR I7=1 TO I2
```

```
101 FOR J7=1 TO J2
102 L7=1+(I7-1)*I(4)
103 M7=1+(J7-1)*J(1)
104 N7=1+(I7-1)*K(4)+(J7-1)*K(1)
105 C(N7)=0
106 FOR K7=1 TO K2
107 C(N7)=C(N7)+A(L7)*B(M7)
108 L7=L7+L2
109 M7=M7+M2
110 NEXT K7
111 NEXT J7
112 NEXT I7
113 EXIT SUB
114 I2=I(2)
115 J2=I(3)
116 I1=I(4)
117 J1=I(1)
118 FOR I7=1 TO I2
119 R(I7)=0
120 NEXT I7
121 B(1)=1
122 FOR K7=1 TO I2
123 L7=0
124 L7=L7+1
125 N7=R(L7)
126 IF (N7<>0) THEN 124
127 J7=L7
128 IF (J7=I2) THEN 152
129 J7=J7+1
130 N7=R(J7)
131 IF (N7<>0) THEN 128
132 S7=0
133 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
134 N7=1+(J7-1)*I(1)+(K7-1)*I(4)
135 P1=ABS(A(M7))
136 P2=ABS(A(N7))
137 M7=1+(L7-1)*I(1)
138 N7=1+(J7-1)*I(1)
139 FOR I7=1 TO I2
140 Q1=ABS(A(M7))
141 Q2=ABS(A(N7))
142 D=P1*Q2-P2*Q1
143 E=P1*Q2+P2*Q1
144 IF (E=0) THEN 146
145 S7=S7+D/E
146 M7=M7+I1
147 N7=N7+I1
148 NEXT I7
149 IF (S7>=0) THEN 128
150 L7=J7
```

```
151 GOTO 128
152 R(L7)=K7
153 S(K7)=L7
154 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
155 B(1)=B(1)*A(M7)
156 IF (B(1)=0) THEN 219
157 FOR I7=1 TO I2
158 IF (I7=K7) THEN 171
159 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
160 N7=1+(L7-1)*I(1)+(I7-1)*I(4)
161 Q=A(N7)/A(M7)
162 M7=1+(K7-1)*I(4)
163 N7=1+(I7-1)*I(4)
164 FOR J7=1 TO J2
165 A(N7)=A(N7)-Q*A(M7)
166 M7=M7+J1
167 N7=N7+J1
168 NEXT J7
169 N7=1+(L7-1)*I(1)+(I7-1)*I(4)
170 A(N7)=-Q
171 NEXT I7
172 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
173 Q=1/A(M7)
174 N7=1+(K7-1)*I(4)
175 FOR J7=1 TO J2
176 A(N7)=Q*A(N7)
177 N7=N7+1
178 NEXT J7
179 A(M7)=Q
180 NEXT K7
181 FOR I7=1 TO I2
182 FOR K7=I7 TO I2
183 L7=R(K7)
184 IF (L7=I7) THEN 186
185 NEXT K7
186 L7=R(I7)
187 M7=1+(I7-1)*I(4)
188 N7=1+(L7-1)*I(4)
189 FOR J7=1 TO J2
190 Q=A(M7)
191 A(M7)=A(N7)
192 A(N7)=Q
193 M7=M7+J1
194 N7=N7+J1
195 NEXT J7
196 R(I7)=I7
197 R(K7)=L7
198 NEXT I7
199 FOR J7=1 TO J2
200 FOR L7=J7 TO J2
```

```
201 K7=S(L7)
202 IF (K7=J7) THEN 204
203 NEXT L7
204 K7=S(J7)
205 M7=1+(J7-1)*I(1)
206 N7=1+(K7-1)*I(1)
207 FOR I7=1 TO I2
208 Q=A(M7)
209 A(M7)=A(N7)
210 A(N7)=Q
211 M7=M7+I1
212 N7=N7+I1
213 NEXT I7
214 IF (J7<>K7) THEN B(1)=-B(1)
215 S(J7)=J7
216 S(L7)=K7
217 NEXT J7
218 EXIT SUB
219 PRINT "STOP BY MTRX FOR ZERO PIVOT"
220 STOP
221 END SUB
222 OPTION NOLET
223 DIM A(16),B(16),C(16)
224 DIM I(4),J(4),K(4)
225 A(1)=0
226 A(2)=0
227 A(3)=0
228 A(4)=1
229 A(5)=0
230 A(6)=0
231 A(7)=1
232 A(8)=0
233 A(9)=0
234 A(10)=1
235 A(11)=0
236 A(12)=0
237 A(13)=1
238 A(14)=0
239 A(15)=0
240 A(16)=0
241 I(1)=1
242 I(2)=4
243 I(3)=4
244 I(4)=4
245 B(1)=0
246 B(2)=0
247 B(3)=0
248 B(4)=0
249 B(5)=0
250 B(6)=0
```

```
251 B(7)=0
252 B(8)=0
253 B(9)=0
254 B(10)=0
255 B(11)=0
256 B(12)=0
257 B(13)=0
258 B(14)=0
259 B(15)=0
260 B(16)=0
261 J(1)=1
262 J(2)=4
263 J(3)=4
264 J(4)=4
265 C(1)=0
266 C(2)=0
267 C(3)=0
268 C(4)=0
269 C(5)=0
270 C(6)=0
271 C(7)=0
272 C(8)=0
273 C(9)=0
274 C(10)=0
275 C(11)=0
276 C(12)=0
277 C(13)=0
278 C(14)=0
279 C(15)=0
280 C(16)=0
281 K(1)=1
282 K(2)=4
283 K(3)=4
284 K(4)=4
285 M=0
286 CALL MTRX (M,A(),I(),B(),I(),C(),I())
287 M=5
288 CALL MTRX (M,A(),I(),C(),I(),C(),I())
289 M=4
290 CALL MTRX (M,A(),I(),B(),I(),C(),I())
291 FOR N=1 TO 16
292 PRINT USING "+#.##########^^^^": 7+C(N)-7
293 NEXT N
294 END
```

```
1 SUB DTMX (Q,A(),I(),D)
2 !**********************************
3 !SUBROUTINE FOR MATRIX DETERMINANT
4 !**********************************
5  !Q = ARGUMENT OF MATRIX A-QI
6  !A = ADDRESS OF MATRIX A
7  !I = SPECIFICATION OF MATRIX A
8  !D = DETERMINANT OF MATRIX A-QI
9  !I(1) = INTERVAL BETWEEN COLUMNS
10 !I(2) = NUMBER OF ROWS
11 !I(3) = NUMBER OF COLUMNS
12 !I(4) = INTERVAL BETWEEN ROWS
13 OPTION NOLET
14 IF (H=1) THEN 17
15 H=1
16 DIM R(256),S(256)
17 I2=I(2)
18 J2=I(3)
19 I1=I(4)
20 J1=I(1)
21 K7=0
22 FOR I7=1 TO I2
23 K7=1+(I7-1)*(I1+J1)
24 A(K7)=A(K7)-Q
25 R(I7)=0
26 NEXT I7
27 D=1
28 FOR K7=1 TO I2
29 L7=0
30 L7=L7+1
31 N7=R(L7)
32 IF (N7<>0) THEN 30
33 J7=L7
34 IF (J7=I2) THEN 58
35 J7=J7+1
36 N7=R(J7)
37 IF (N7<>0) THEN 34
38 S7=0
39 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
40 N7=1+(J7-1)*I(1)+(K7-1)*I(4)
41 P1=ABS(A(M7))
42 P2=ABS(A(N7))
43 M7=1+(L7-1)*I(1)
44 N7=1+(J7-1)*I(1)
45 FOR I7=1 TO I2
46 Q1=ABS(A(M7))
47 Q2=ABS(A(N7))
48 E=P1*Q2-P2*Q1
49 F=P1*Q2+P2*Q1
50 IF (F=0) THEN 52
```

```
51 S7=S7+E/F
52 M7=M7+I1
53 N7=N7+I1
54 NEXT I7
55 IF (S7>=0) THEN 34
56 L7=J7
57 GOTO 34
58 R(L7)=K7
59 S(K7)=L7
60 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
61 D=D*A(M7)
62 IF (D=0) THEN 125
63 FOR I7=1 TO I2
64 IF (I7=K7) THEN 77
65 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
66 N7=1+(L7-1)*I(1)+(I7-1)*I(4)
67 P=A(N7)/A(M7)
68 M7=1+(K7-1)*I(4)
69 N7=1+(I7-1)*I(4)
70 FOR J7=1 TO J2
71 A(N7)=A(N7)-P*A(M7)
72 M7=M7+J1
73 N7=N7+J1
74 NEXT J7
75 N7=1+(L7-1)*I(1)+(I7-1)*I(4)
76 A(N7)=-P
77 NEXT I7
78 M7=1+(L7-1)*I(1)+(K7-1)*I(4)
79 P=1/A(M7)
80 N7=1+(K7-1)*I(4)
81 FOR J7=1 TO J2
82 A(N7)=P*A(N7)
83 N7=N7+J1
84 NEXT J7
85 A(M7)=P
86 NEXT K7
87 FOR I7=1 TO I2
88 FOR K7=I7 TO I2
89 L7=R(K7)
90 IF (L7=I7) THEN 92
91 NEXT K7
92 L7=R(I7)
93 M7=1+(I7-1)*I(4)
94 N7=1+(L7-1)*I(4)
95 FOR J7=1 TO J2
96 P=A(M7)
97 A(M7)=A(N7)
98 A(N7)=P
99  M7=M7+1
100 N7=N7+1
```

DTMX

```
101 NEXT J7
102 R(I7)=I7
103 R(K7)=L7
104 NEXT I7
105 FOR J7=1 TO J2
106 FOR L7=J7 TO J2
107 K7=S(L7)
108 IF (K7=J7) THEN 110
109 NEXT L7
110 K7=S(J7)
111 M7=1+(J7-1)*I(1)
112 N7=1+(K7-1)*I(1)
113 FOR I7=1 TO I2
114 P=A(M7)
115 A(M7)=A(N7)
116 A(N7)=P
117 M7=M7+I1
118 N7=N7+I1
119 NEXT I7
120 IF (J7<>K7) THEN D=-D
121 S(J7)=J7
122 S(L7)=K7
123 NEXT J7
124 EXIT SUB
125 END SUB
126 OPTION NOLET
127 DIM A(16),I(4),C(16)
128 DIM Q0(5),D0(5)
129 A(1)=2
130 A(2)=1
131 A(3)=3
132 A(4)=4
133 A(5)=1
134 A(6)=-3
135 A(7)=1
136 A(8)=5
137 A(9)=3
138 A(10)=1
139 A(11)=6
140 A(12)=-2
141 A(13)=4
142 A(14)=5
143 A(15)=-2
144 A(16)=-1
145 I(1)=1
146 I(2)=4
147 I(3)=4
148 I(4)=4
149 Q0(1)=-1
150 Q0(2)=-.5
```

```
151 Q0(3)=0
152 Q0(4)=+.5
153 Q0(5)=+1
154 FOR K=1 TO 5
155 FOR N=1 TO 16
156 C(N)=A(N)
157 NEXT N
158 Q=Q0(K)
159 CALL DTMX (Q,C(),I(),D)
160 D0(K)=D
161 PRINT USING "+#.##########^^^^": D
162 NEXT K
163 END
```

```
1 SUB RVMX (E,M,R,A(),I(),V())
2 !**********************************
3 !CHARACTERISTIC VECTOR OF A MATRIX
4 !**********************************
5 !M = MODE OF OPERATION
6 !R = ADDRESS OF ROOT
7 !A = ADDRESS OF MATRIX A
8 !I = SPECIFICATION OF MATRIX A
9 !V = ADDRESS OF VECTOR
10 !I(1) = INTERVAL BETWEEN COLUMNS
11 !I(2) = NUMBER OF ROWS
12 !I(3) = NUMBER OF COLUMNS
13 !I(4) = INTERVAL BETWEEN ROWS
14 !M = 0 FOR INITIALIZATION
15 !M = 1 FOR CONTINUATION
16 !
17 OPTION NOLET
18 IF (H=1) THEN 23
19 I2=I(2)
20 J2=I(3)
21 I1=I(4)
22 J1=I(1)
23 IF (M<>0) THEN 78
24 FOR I7=1 TO I2
25 K7=1+(I7-1)*(I1+J1)
26 A(K7)=A(K7)-R
27 NEXT I7
28 I3=I2-1
29 IF I3>0 THEN 32
30 V(1)=1
31 EXIT SUB
32 FOR K7=1 TO I3
33 L7=K7+1
34 FOR I7=L7 TO I2
35 S7=0
36 M7=1+(K7-1)*I(1)+(K7-1)*I(4)
37 N7=1+(K7-1)*I(1)+(I7-1)*I(4)
38 P1=ABS(A(M7))
39 P2=ABS(A(N7))
40 IF (P2=0) THEN 63
41 FOR J7=K7 TO J2
42 M7=1+(J7-1)*I(1)+(K7-1)*I(4)
43 N7=1+(J7-1)*I(1)+(I7-1)*I(4)
44 Q1=ABS(A(M7))
45 Q2=ABS(A(N7))
46 E7=P1*Q2-P2*Q1
47 F7=P1*Q2+P2*Q1
48 IF (F7=0) THEN 50
49 S7=S7+E7/F7
50 M7=M7+J1
```

```
51 N7=N7+J1
52 NEXT J7
53 IF (S7>=0) THEN 63
54 M7=1+(K7-1)*I(1)+(K7-1)*I(4)
55 N7=1+(K7-1)*I(1)+(I7-1)*I(4)
56 FOR J7=K7 TO J2
57 P=A(M7)
58 A(M7)=A(N7)
59 A(N7)=P
60 M7=M7+J1
61 N7=N7+J1
62 NEXT J7
63 NEXT I7
64 FOR I7=L7 TO I2
65 M7=1+(K7-1)*I(1)+(K7-1)*I(4)
66 N7=1+(K7-1)*I(1)+(I7-1)*I(4)
67 IF (A(M7)=0) THEN 77
68 P=A(N7)/A(M7)
69 FOR J7=K7 TO J2
70 M7=1+(J7-1)*I(1)+(K7-1)*I(4)
71 N7=1+(J7-1)*I(1)+(I7-1)*I(4)
72 A(N7)=A(N7)-P*A(M7)
73 M7=M7+J1
74 N7=N7+J1
75 NEXT J7
76 NEXT I7
77 NEXT K7
78 Q=0
79 FOR J7=1 TO J2
80 V(J7)=0
81 NEXT J7
82 N7=1+(J2-1)*I(1)+(I2-1)*I(4)
83 IF (ABS(A(N7))>E) THEN 87
84 Q=1
85 V(I2)=1
86 A(N7)=1
87 FOR K7=1 TO I3
88 I7=I2-K7
89 L7=I7+1
90 S7=0
91 FOR J7=L7 TO J2
92 M7=1+(J7-1)*I(1)+(I7-1)*I(4)
93 S7=S7+A(M7)*V(J7)
94 NEXT J7
95 N7=1+(I7-1)*(I1+J1)
96 IF (ABS(A(N7))>E) THEN 103
97 IF (Q=1) THEN 104
98 IF (S7<>0) THEN 106
99 Q=1
100 V(I7)=1
```

```
101 A(N7)=1
102 GOTO 104
103 V(I7)=-S7/A(N7)
104 NEXT K7
105 EXIT SUB
106 FOR J7=1 TO J2
107 V(J7)=0
108 NEXT J7
109 EXIT SUB
110 END SUB
111 OPTION NOLET
112 DIM A(16),I(4),V(4)
113 A(1)=0
114 A(2)=0
115 A(3)=0
116 A(4)=1
117 A(5)=0
118 A(6)=0
119 A(7)=1
120 A(8)=0
121 A(9)=0
122 A(10)=1
123 A(11)=0
124 A(12)=0
125 A(13)=1
126 A(14)=0
127 A(15)=0
128 A(16)=0
129 I(1)=1
130 I(2)=4
131 I(3)=4
132 I(4)=4
133 E=.000001
134 M=0
135 R=+1
136 CALL RVMX (E,M,R,A(),I(),V())
137 FOR N=1 TO 4
138 PRINT USING "+#.###########^^^^": V(N)
139 NEXT N
140 M=1
141 R=+1
142 CALL RVMX (E,M,R,A(),I(),V())
143 FOR N=1 TO 4
144 PRINT USING "+#.###########^^^^": V(N)
145 NEXT N
146 END
```

DISTRIBUTION

Defense Documentation Center 2
Cameron Station
Alexandria, VA 22314

Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943

Research Administration, Code 09 1
Naval Postgraduate School
Monterey, CA 93943

Mathematics Department, Code MA 1
Naval Postgraduate School
Monterey, CA 93943

Computer Center, Code 51 1
Naval Postgraduate School
Monterey, CA 93943

Professor T. E. Kurtz 1
True BASIC Inc.
12 Commerce Ave.
West Lebanon, NH 03784-1669

Dr. A. V. Hershey, Code 51 7
Naval Postgraduate School
Monterey, CA 93943